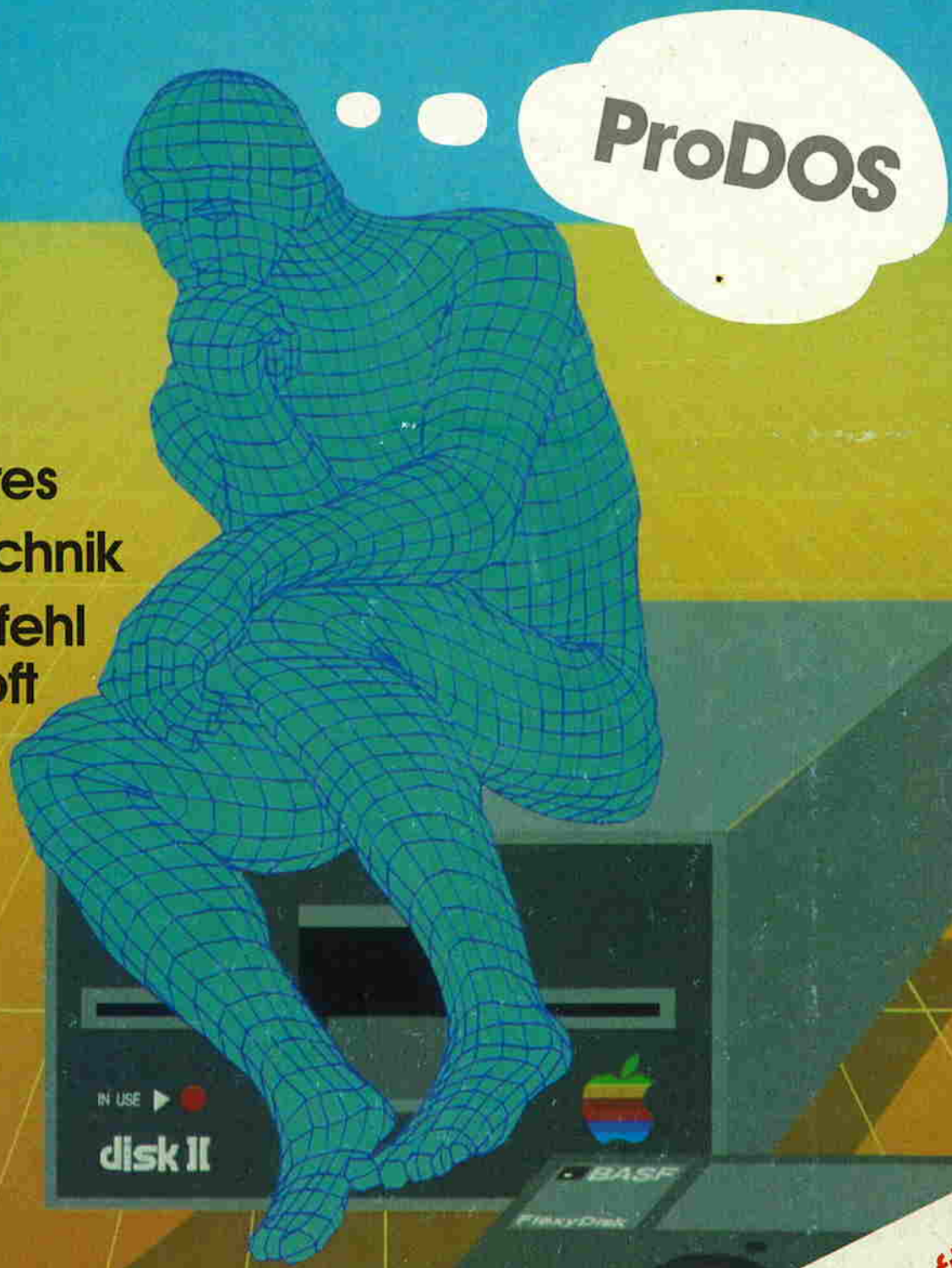
Dezember 84/Januar 85 DM 6,50

# PEEKER

## MAGAZIN FÜR APPLE-COMPUTER




**2/84**

ProDos für Anfänger

Double Hires

Modemtechnik

Instring-Befehl in Applesoft

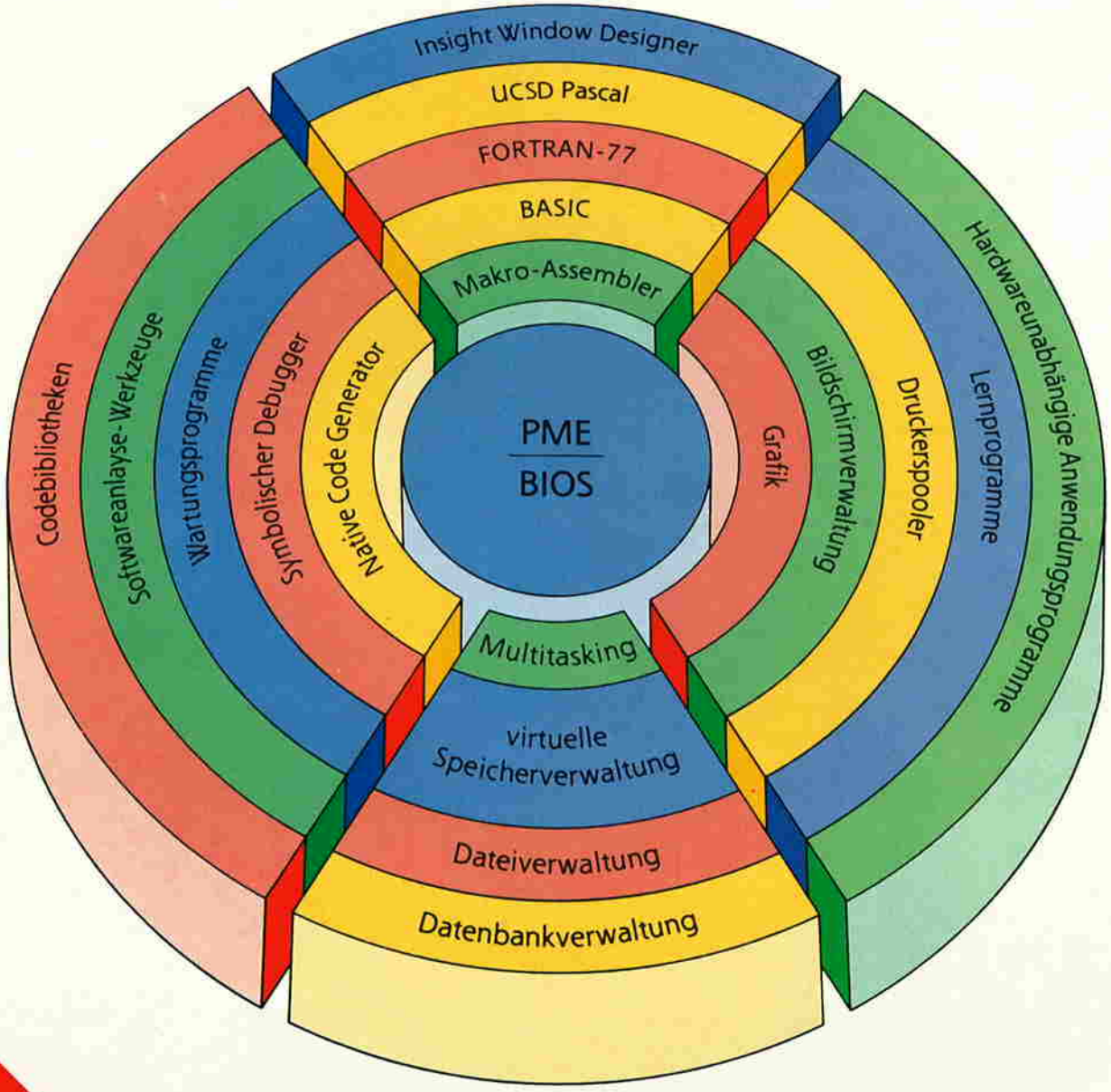
Ultraterm-Karte

65CO2C-Rechner

Neue Serie Microsoft Basic für Macintosh

Hüthig PUBLIKATION

# Das UCSD p-System.




**Jetzt auch für Apple Macintosh**

FOCUS
Computer GmbH
Friesenstrasse 14 · 3000 Hannover 1

# EDITORIAL

Um es gleich vorweg zu sagen: Die Glückszahl in dem Gewinnspiel war 45. Diese Zahl hatten wir gewählt, weil der „Peeker" unsere 45. Fachzeitschrift ist. Von den mehr als 30.000 Empfängern des Heftes 1/84 gingen über 4.000 Gewinn-Karten mit Angaben zum Gerätebesitz usw. ein, die wir inzwischen statistisch ausgewertet haben. Obgleich die Stichprobe von 13.3% (4.000 von 30.000) nicht 100% repräsentativ ist, denn ein Techniker oder Kaufmann ist wahrscheinlich weniger geneigt, sich an einem Gewinnspiel zu beteiligen, als ein Student oder Schüler, kommt der Erhebung doch eine hohe statistische Signifikanz zu. Dabei kamen einige Überraschungen zutage, die zugleich bestätigen, daß wir mit unserem „Peeker" den richtigen Kurs eingeschlagen haben:

– Apple II+, Apple IIe/IIc und Apple-Kompatible halten sich die Waage, d.h. es gibt genausoviele Apple II+, wie es Apple-Kompatible gibt, und die Zahl der Apple-Kompatiblen entspricht der Gesamtsumme aus Apple IIe und IIc. Daraus folgt, daß wir auch den Kompatiblen gebührende Aufmerksamkeit schenken müssen.

– Der IIc ist für sich genommen noch sehr schwach repräsentiert, und er dürfte den IIe wohl kaum überrunden, da die meisten Benutzer offenbar mehr Wert auf Erweiterungsfähigkeit denn auf Kompaktheit legen.

– Nicht verwunderlich ist die „Fast-Null-Verbreitung" vom Apple III und von der Lisa. Der Apple III hat bereits seine letzte Ruhestatt gefunden, und die Lisa folgt zaudernden Schrittes hinterdrein. Konkret gesprochen, wurde die Produktion des Apple III, den wir auf unserer Gewinn-Karte nicht mehr erwähnt hatten, bereits still und heimlich eingestellt, und wenn nicht alle Zeichen trügen, dürfte die Lisa, sobald der Macintosh seine Kinderschuhe ausgetreten hat, den gleichen Weg gehen, denn sie hat seit der Markteinführung vor anderthalb Jahren nie Tritt fassen können, und zwar weniger wegen der Qualität dieses Gerätes, sondern wegen des exorbitant hohen Preises.

– Bedenkt man, daß allein auf der Apple-Expo in Köln, die ausschließlich im Zeichen des Macintosh stand, über 3.000 „Peeker"-Hefte an Besucher gegen Schutzgebühr verteilt wurden, so enttäuscht der ungewöhnlich geringe Marktanteil des Macintosh mit weniger als 1 % aller Apple-Geräte. Ungeachtet dessen werden wir im „Peeker" auch den Macintosh berücksichtigen, doch wäre es sträflich, die übrigen 99% der Apple-Besitzer zu vernachlässigen. Wir haben uns zunächst überlegt, ob wir das katastrophale Abschneiden des Macintosh, das sich übrigens mit einer Reihe von Anrufen frustrierter Käufer deckt, publik machen sollten, da der Macintosh in „Apple's unabhängigem Computermagazin" unter Headlines wie „92.000 Macintosh verkauft" oder „Macintosh holt auf" in den Himmel gepriesen wird. Hierzu muß man jedoch wissen, daß das erste Heft von „Apple's" von der Firma Apple höchstpersönlich an die Apple-Händler versandt worden ist. Was es in diesem Zusammenhang mit dem Wort „unabhängig" im Titel der „Apple's" auf sich hat, mag sich der Leser selbst denken. Wir haben es gottlob nicht nötig, uns lautstark als unabhängig zu bezeichnen. Wir sind es ganz einfach! Und deshalb wäre es eine Irreführung der Leser, wenn wir von Abertausenden glücklicher Macintosh-Besitzer berichten würden. Beispielsweise werden wir bei Hard- und Software-Tests stets fair sein und das, was Hand und Fuß hat, auch sehr positiv rezensieren (siehe UltraTerm in diesem Heft). Aber wollen Sie, daß wir umgekehrt die Schattenseiten zu erwähnen „vergessen"? Ihre Stellungnahme zu diesem Thema würde uns sehr interessieren.

**Ulrich Stiehl**

Hüthig
PUBLIKATION

# inhalt

# 2/84

**Impressum**
Peeker
Magazin für Apple-Computer
1. Jahrgang 1984
ISSN 0176–9200
© für den gesamten Inhalt
einschließlich der Programme
Dr. Alfred Hüthig Verlag,
Heidelberg 1984

Verleger und Herausgeber:
Dipl.-Kfm. Holger Hüthig
Geschäftsführung Zeitschriften:
Heinz Melcher
Chefredakteur:
Ulrich Stiehl (us) Tel. (0 62 21) 48 93 52

Anzeigenleitung:
Bernd Beutel, Tel. (0 62 21) 48 92 18
z. Zt. gilt Anzeigenpreisliste Nr. 1
Vertriebsleitung:
Ruth Biller, Tel. (0 62 21) 48 92 80
Produktionsleitung: Gunter Sokollek
Gestaltung: Rainer Schmitt

# peeker


Editorial . . . . . 5
Impressum . . . . . 6
Leserbriefe . . . . . 8
Händlerprofil . . . . . 69
Gewinner . . . . . 70
Hinweis für Autoren . . . . 72
Neue Produkte . . . . . 74
Neue Bücher . . . . . 76
Vorschau . . . . . 80



## 12
**ProDOS für Anfänger**
**Teil 1: Was ist ProDOS?**

In diesem ersten Teil einer umfangreichen Serie über ProDOS für Applesoft-Programmierer werden die Vor- und Nachteile dieses neuen Betriebssystems gegenüber DOS 3.3 herausgearbeitet.

## 20
**Was Geschäftsleute vom ProDOS-FILER wissen sollten**

## 24
**Wie man die Grafik verdoppelt**
**Teil 2: Double Hires**

In Fortsetzung unserer Doppel-Grafik-Serie werden diesmal die Geheimnisse der doppelt hochauflösenden Grafik des IIe/IIc mit 560 x 192 Bildpunkten gelüftet.

## 36
**Modems kommen in Mode**
**Praktische Ratschläge für die Telekommunikation mit Modem**

## 40
**Ampersand macht's möglich**
**INSTRING-Befehl in Applesoft**

Wie man das in Applesoft fehlende If-Then-Else durch On-Goto und andere Konstruktionen simulieren kann und was es mit den Booleschen Operatoren auf sich hat, wird in einer für Anfänger leicht verständlichen Form erläutert.

## 48
**IF-THEN-ELSE-Befehl simuliert in Applesoft**

## 51
**Wie man Arrays löscht**

## 54
**Microsoft Basic leicht geMACht**
**Teil 1: Operanden und Operatoren**

In diesem ersten Teil unserer umfangreichen Serie über das Macintosh-Basic von Microsoft wird das definitorische Grundgerüst dieses neuen Basic-Dialekts vermittelt.

## 60
**Gestochen scharf**
**Die Videxkarte UltraTerm für den Apple II**

## 64
**Prometric**
**Ein Apple-Kompatibler mit 65C02C**

## 66
**S.A.M.**
**The Software Automatic Mouth oder wie man den Apple zum Sprechen bringt**



Verlag:
Dr. Alfred Hüthig Verlag GmbH
Im Weiher 10, Postfach 10 28 69
6900 Heidelberg
Telefon (0 62 21) 4 89-1
Telex 4-6 17 27 hued d.

Erscheinungsweise:
1984 2 Hefte, ab 1985 12 Hefte jährlich.
Jahresabonnement DM 58,–, einschließlich MwSt, im Inland portofrei.
Einzelheft DM 6,50

**Zahlungen:** an den Dr. Alfred Hüthig-Verlag GmbH, D-6900 Heidelberg 1: **Postscheckkonten:** BRD: Karlsruhe 48545-753; Österreich: Wien 755588; Schweiz: Basel 40-24417; Niederlande: Den Haag 145728; Italien: Mailand 47718; Belgien: Brüssel 723026; Dänemark: Kopenhagen 34969; Norwegen: Oslo 99424; Schweden: Stockholm 547776-5

**Bankkonten:** Landeszentralbank Heidelberg 67207341; BLZ 67200000; Deutsche Bank Heidelberg 0 2165 041; BLZ 67270003; Bezirkssparkasse Heidelberg 20451, BLZ 67250020.

Herstellung: Heidelberger Verlagsanstalt
Printed in Germany

**65C02**

Zuerst einmal möchte ich Ihnen zu dieser neuen Zeitschrift gratulieren. Es wurde allmählich Zeit, daß sich im deutschen Sprachraum eine Zeitschrift ganz dem Apple II widmete, nachdem die Blättchen für Commodore, Sinclair und ähnliche Geräte wie Pilze aus dem Boden schossen. Es entstand langsam der Eindruck, daß diese Marken die einzigen wären und der „namenlose" Rest nichts tauge.

Bisher gab es in bezug auf den Apple nur das Blatt der A.U.G.E., aber in letzter Zeit wurde es (bzw. die Mitgliedschaft) immer teurer und der Inhalt immer magerer. Ich selbst bin noch bis Ende dieses Jahres Mitglied (falls es Sie interessiert: 4 Jahre Mitglied, Nummer 293), im Verlauf dieses Jahres wurde der Wasserkopf in der Verwaltung immer größer, die Leistungen immer weniger.

Nun wieder zu Ihrer Zeitschrift, also wie gesagt ist die erste Nummer sehr gut, bis auf ein paar Kleinigkeiten, aber die kann man ja verbessern.

In Ihrem Artikel über die CMOS-CPU's erwähnen Sie unverständlicherweise nicht den ersten Hersteller, der die 65C02 auf den Markt brachte, GTE mit der Version 65SC02. Mit dieser CPU hat es eine besondere Bewandtnis. Sie läuft im Gegensatz zur R65C02 von Rockwell anstandslos im Apple II europlus und den kompatiblen Nachbauten. Sie hat den gleichen Befehlssatz wie die CPU von NCR, also keine Einzelbitbefehle (BBR, BBS, RMB, SMB) auf den Zero-Page-Adressen. Wie erwähnt, läuft die R65C02 nur auf dem IIe, natürlich funktioniert hier auch die 65SC02.

Dieser Umstand wurde bisher noch nirgends erwähnt, und seine Bekanntmachung hätte sicher manchem Applebesitzer eine Enttäuschung erspart.

Ansonsten hoffe ich, daß Sie mit Ihrer Zeitschrift die Besitzer der „alten" Geräte nicht im Stich lassen werden, denn der II plus hat gegenüber dem neuen IIe immer noch einige Vorteile.

*Thomas Holzwarth, Ludwigsburg*

**Apple IIe mit A-Motherboard**

Ihren Bericht über die doppelte Loresgrafik in Ihrem ersten Heft möchte ich in zwei Punkten korrigieren:

1. Es wurden auch in Westdeutschland Apple IIe mit einer Platine der Version A ausgeliefert. Ich selbst besitze ein solches, anscheinend exotisches Exemplar, das ich im April 1983 bei einem autorisierten Applehändler gekauft habe.

2. Auch mit meiner A-Version ist nach dem Aufstecken des Jumperblocks die doppelte Lores- und Hiresgrafik möglich, ohne daß mein IIe bisher abgestürzt wäre. Dabei muß man keine anderen Peeks und Pokes verwenden, sondern kann mit den bei Ihnen und auch im Manual für die erweiterte 80-Zeichenkarte angegebenen Adressen arbeiten.

Vor meinen Versuchen mit der erweiterten 80-Zeichenkarte fragte ich bei Apple in München an, ob die doppelte Auflösung mit der A-Version möglich sei, was rundweg verneint wurde, wobei sie natürlich auch die Änderungen, die bei der B-Version vorgenommen wurden, nicht verraten wollten. Mich würde jetzt interessieren, welche Abweichungen zwischen der A- und der B-Version wirklich bestehen.

*Dieter Sabath, Vilsbiburg*


**MMU 2.0**

**Memory Managements Utilities**

**für die Apple IIe 64K-Karte**
**DOS 3.3 (und ProDOS)**

**von U. Stiehl**

1984, Diskette und Manual, DM 98,–
ISBN 3-7787-1023-1

Insgesamt enthält die neue „MMU 2.0"-Diskette über 25 Programme, die neue Einsatzmöglichkeiten für die Extended 80 Column Card (erweiterte 80-Z-Karte = 64K-Karte für den Apple IIe) erschließen. Ein Teil der Programme laufen auch auf dem Apple II Plus, doch ist „MMU 2.0" primär für 64K-Karte-Besitzer gedacht.

Gerätevoraussetzung: Apple IIe mit 64K-Karte oder IIc

**Hüthig Software Service,**
**Postfach 10 28 69, D-6900 Heidelberg**


**65C02**

Als neues Mitglied Ihrer Lesergemeinschaft möchte ich mich auch gleich ein wenig zu dem vor mir liegenden Heft äußern und ein paar Vorschläge für spätere Ausgaben machen.

Zuerst einmal die Aufmachung: Sehr schön! Ein klares, sauberes und auch farbenfrohes Gesicht. Man sieht schon auf den ersten Blick, für wen dies gedacht ist. Das habe ich bei mancher anderen Zeitschrift vermißt, wo erst das 173. Programm für den „XYZ 2001" darüber informierte, daß ich hier wohl falsch war...

Zum Inhalt: Ein Silberstreif am Horizont! Da ist zum Beispiel das „65C02-Feature". Endlich einmal klare Aussagen. Welcher (Hardware-)Laie traut sich denn schon, den Prozessor auszutauschen? Ich habe zwar durchaus schon Erfahrungen auf dem Gebiet der Digitalelektronik sammeln können, aber an das Herz meines Apple hätte ich mich nie getraut aus lauter Angst, mein „Emil" könnte sich auf DFÜ spezialisieren (Indianer-Methode: 3 lange und ein kurzes Rauchzeichen = Ende!). Und die Beschreibung der „Accelerator IIe" war auch einmal etwas Brauchbares. Nach so einem Artikel kann man sich doch endlich vorstellen, was einen erwartet, sollte man sich dieses Gerät kaufen.

*Franz Offenbächer, München*

**„Literaturkatalog"**

Der Großteil der Beiträge hat mir gut gefallen, insbesondere der Artikel über den neuen 65C02-Prozessor und der Double-Resolution-Graphic-Beitrag.

Sehr schön ist, daß nicht auch diese Zeitschrift bei „Null" anfängt, d.h. mit einem BASIC-Grundkurs beginnt, so wie das wohl in 90% der Computerzeitschriften sonst der Fall ist. Bei einem Apple-Besitzer darf man ja auch gewisse BASIC-Kenntnisse voraussetzen.

Wenn man die Zeitschrift durchsieht, so hat man fast den Eindruck, einen Software-und Literaturkatalog einer Firma namens U. Stiehl in Händen zu halten. Ich habe einmal nachgezählt: Ihr Name erscheint allein in Werbeanzeigen für Software und Bücher auf nicht weniger als 13 Seiten! Für die (zugegebenermaßen nicht schlechten) Bücher „DOS 3.3" und „ProDOS" wird fünfmal geworben.

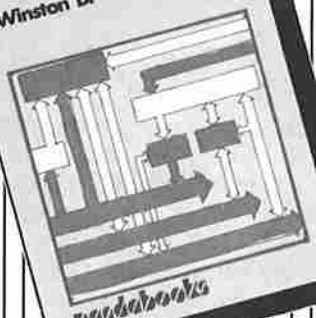
*Stefan Stieler, Gießen*

Ihr Einwand ist berechtigt, doch sollte zunächst einmal dokumentiert werden, daß „Peeker" redaktionell von jemandem betreut wird, der den Apple in- und auswendig kennt, was keineswegs selbstver-

# pandabooks

## für den Computer-Fuchs

**Apple II Schaltpläne**
**Winston D. Gayler**



pandabooks

Eine detaillierte Beschreibung der Apple II-Schaltungen.
Wenn Sie Ihren Apple selbst reparieren, Interface-Karten oder Schaltungserweiterungen entwerfen oder einfach nur besser über das Innenleben Ihres Apples Bescheid wissen wollen – dieses Buch bietet Ihnen eine Fülle an Informationen: Schaltpläne und Zeitdiagramme, Theorie und praktische Tips.

ISBN: 3-89058-012-2 ..... DM 64,--

**Apple Pascal Grafik**
**Tom Swan**

pandabooks

22 PASCAL-Programme, mit denen Sie die Grafik-Möglichkeiten Ihres Apples voll ausschöpfen:
DESIGNER ermöglicht es Ihnen, eigene Zeichensätze zu entwerfen.
GREDIT unterstützt Sie beim Entwerfen und Abspeichern kompletter Bildschirmgrafiken.
PRINTFOTO bringt Ihre Entwürfe aufs Papier.
Darüberhinaus bietet das Buch eine Fülle fertiger Prozeduren, die Sie zeitsparend in Ihre eigenen Programme einbauen können.

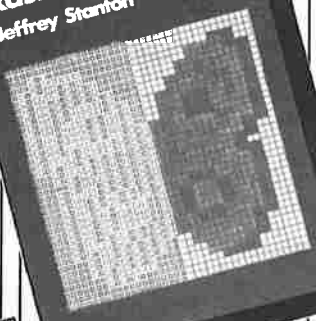
ISBN: 3-89058-009-2 ..... DM 49,--
komplett mit Disk DM 89,--

Endlich anspruchsvolle Computergrafik für BASIC-Programmierer!
Mikrocomputer Grafik
- enthält fast 80 lauffertige BASIC-Programme, die die beschriebenen Grafikkonzepte illustrieren
- ist für den Anfänger geschrieben und erfordert nur elementare Mathematikkenntnisse
- enthält Techniken zur Erzeugung dreidimensionaler Grafiken in zwei Dimensionen, Hidden Line- und Hidden Surface-Methoden und Skalierung, Rotation und Translation von Grafiken
- bietet eine Einführung in die Animationstechnik

ISBN: 3-89058-000-9 ..... DM 49,--
komplett mit Disk DM 89,--

**Mikrocomputer Grafik**

**Apple II Raster Grafik**
**Jeffrey Stanton**



Die Qualität kommerzieller Arcadespiele läßt sich mit APPLESOFT BASIC alleine nicht erreichen. Jeffrey Stanton führt in die Eigenarten der hochauflösenden Apple-Grafik ein und präsentiert schließlich eine Reihe extrem schneller Assembler-Routinen, mit denen Sie viele Effekte bekannter Spiele selbst programmieren können. Gute BASIC-Kenntnisse werden vorausgesetzt, eine Einführung in Assemblerprogrammierung wird gegeben.

ISBN: 3-89058-006-8 ..... DM 49,--
komplett mit Disk DM 89,--

**Apple II Assembler-Programmierung**
**Roger Wagner**

pandabooks

Das Assembler-Lehrbuch für BASIC-Kenner. Roger Wagner, der Autor vieler bekannter Software-Pakete, schrieb eine monatliche Kolumne über Assembler-Programmierung in der APPLE-Zeitschrift SOFTALK.
Der vorliegende Band faßt diese Reihe, korrigiert und erweitert, zusammen:
Eine stufenweise Einführung in die Befehle und Strukturen der Assemblersprache, mit vielen Beispielen von der einfachen Tongenerierung bis zum Diskettenzugriff.

ISBN: 3-89058-003-3 ..... DM 48,--
komplett mit Disk DM 89,--

Unsere Bücher gibt es bei Ihrem Apple-Händler, in vielen Buchhandlungen oder direkt bei:
Pandabooks
Bismarckstr. 67 1000 Berlin 12
Tel.: (030) 342 88 00

ständlich ist. Es gibt bekanntlich Apple-Zeitschriften, deren Redakteure noch nie ein größeres Assemblerprogramm geschrieben haben und die deshalb selbst die einfachsten Dinge durcheinanderbringen. Wir haben uns für „Peeker" Gründlichkeit und Akribie auf die Fahne geschrieben, weil damit dem Leser wohl am besten gedient ist. Anm. der Red.

**Accelerator**

Ich begrüße es sehr, daß endlich einmal eine „nur Apple"-Zeitung auf dem deutschen Markt erschienen ist, da entsprechende Publikationen in Englisch ca. das Dreifache Ihrer Zeitschrift kosten und Kontakt über die Zeitschrift mit anderen Usern oder Anbietern nahezu unmöglich ist.
Beim ersten Griff nach der Zeitschrift fällt auf, daß diese noch erfreulich dünn ist, da nur wenig Werbung und keine Leserkontaktseiten etc. enthalten sind, was sich aber wahrscheinlich bald ändern wird.
Was sich nicht ändern sollte, ist das klare Erscheinungsbild der Artikel, so besonders die großzügige Darstellung der Assemblerlistings, die nicht, wie in vielen anderen Zeitschriften, als riesiger Hex-Dump auf eine Seite geklatscht wurden. Auch die Beschreibungen z.B. des 65C02-Prozessors oder der Accelerator-Karte lassen an Übersichtlichkeit und vor allem vom Inhalt her kaum noch Wünsche offen, denn meistens findet man bei Produktbeschreibungen nur eine Übersicht der Features und kurze Anmerkungen über die Bedienung.
Software-Themen wie das ProDOS-Patch für geänderte F8-ROMs sind nicht nur interessant, sondern bieten auch echten Nutzwert für viele Leser, die Kleinschrift nachgerüstet haben oder, so wie ich, einen Apple-Nachbau besitzen.
Dagegen dürfte der Patch für ein 80-Track-Laufwerk nur die wenigsten Leser interessieren, obwohl natürlich ProDOS erst ab dieser Laufwerkgröße interessant wird, wenn man nicht gleich zur Festplatte greift.

Der Artikel über Turbo-Pascal enthält einen wesentlichen Fehler. Ihr Autor unterliegt dem Irrtum, daß der Wunsch nach Pascal wegen der umständlichen Systemhandhabung unerfüllt bleibt. Mitnichten! Es liegt vielmehr an dem für Hobby-Anwender kaum erschwinglichen Preis von Apple-Pascal, was aber nach dem Erscheinen von Turbo-Pascal gegenstandslos geworden ist.
*Michael Preiß, Hochheim*

**Turtle Graphic**

Der Artikel „Applesoft simuliert Turtle Graphic" hat mich ebenfalls fasziniert. Da ich mit Logo schon die ersten Erfahrungen gemacht habe und das einfache Programmieren mit Logo inzwischen schätze, freue ich mich schon auf das Arbeiten mit dem abgedruckten Programm. Dabei hat mich jedoch der Hinweis am Ende des Listings überrascht. Denn zu Beginn meiner Programmierung in Applesoft ist mir beim Ablauf eines mehr als 8K großen Programmes mit Grafik das Programm teilweise gelöscht worden. Dagegen hilft jedoch folgendes kleine Programm:

```
10 Poke 96 * 256,0: Poke 104,96
20 REM Programmvorspann löschen
30 REM Startadresse hochsetzen
40 PRINT CHR$(4);„RUN PROGRAMM-
NAME"
```

das man vor dem entsprechenden Grafikprogramm starten muß.
*Bernd Götz, Hürth*

**Null und O-Buchstabe**

Mir gefällt „Peeker" wirklich sehr gut. Was mir besonders gut gefällt, sind die ausführlichen Beschreibungen von den Programmen.
Aber in den Listings sind die Nullen nicht von den O's zu unterscheiden! Das kann leicht zu Verwirrungen führen (z.B. AO und AO) .Ich würde es sehr begrüßen, wenn zukünftig die Null als „Computer-Null" mit Schrägstrich gedruckt wird.
*Oliver Pulsfort, Schwarzenbek*
Wird ab Heft 1/85 geändert. Anm. der Red.

# Ergänzungen und Berichtigungen

**ProDOS-Patch für geänderte F8-ROMs**
(Heft 1/84, S. 55)

In der zweiten Spalte muß es statt
2659: EA EA (falsch)
heißen
265B: EA EA (richtig)

Man beachte im übrigen, daß der Patch nur für den Apple II Plus und nicht für beliebige Nachbauten gedacht war. Weitergehende Patches für „Kompatible" werden in der nächsten Ausgabe von „Peeker" veröffentlicht. Einen Universal-ProDOS-Patch für Schein-Kompatible aller Art gibt es schon deshalb nicht, weil insbesondere das BASIC.SYSTEM teils illegale Monitor- und Applesoft-Interpreter-Einsprünge benutzt, die bei Schein-Kompatiblen oft nicht vorhanden sind. Für bestimmte Schein-Kompatible ist daher nur ein Individual-ProDOS-Patch möglich.

**Applesoft simuliert Turtle Graphic**

(Heft 1/84, S. 26)
Auf vielfachen Wunsch bringen wir nachfolgend einige weitere Übungsbeispiele zum Turtle-Graphic-Programm:

```
:LEARN> POT 2 PICK SWAP 1 - DO 2
PICK * LOOP SWAP PULL <
:LEARN> QUAD 2 POT <
:LEARN> QLISTE DUP DO DUP QUAD
SWAP 1 - LOOP <
:LEARN> POS PENUP MOVETO PEN-
DOWN <
:LEARN> RAHMEN 0 0 POS 90 TURNTO
4 DO 300 MOVE 90 TURN LOOP <
:LEARN> NECK SWAP DUP 360 SWAP /
SWAP DO DUP TURN SWAP DUP MOVE
SWAP LOOP PULL PULL <
:LEARN> SIEBEN 7 30 NECK <
:LEARN BILD TURTLE RAHMEN 140 80
POS 36 DO SIEBEN 10 TURN LOOP <
```

X Y POT – erhebt X zur Potenz Y und schreibt das Ergebnis auf den Stack
X QUAD – schreibt das Quadrat von X auf den Stack
N Q LISTE – schreibt eine Liste der Quadrate von 0 – N auf den Stack
X Y POS – positioniert die Turtle auf die Stelle X, Y
RAHMEN – zeichnet einen Rahmen um den HGR1-Bildschirm
N X NECK – zeichnet ein N-Eck mit n Ecken und der Seitenlänge 30
SIEBEN – zeichnet ein Siebeneck mit der Seitenlänge 30
BILD – zeichnet ein Bild aus verdrehten Siebenecken

# Die Mikrocomputer-Zeitschrift, die ihre Leser zu Profis macht:

mc liefert Grundlagen für alle, die sich mehr als nur vordergründig mit der Mikrocomputerei befassen möchten…

mc setzt allgemeines technisches Verständnis voraus, weil sie den ernsthaft Interessierten weiterbringen will…

mc informiert umfassend. Über Computer und Peripherie, über Programmiersprachen und Betriebssysteme…

mc testet Hardware und prüft Programme. mc gibt so Entscheidungshilfe vor einer Anschaffung.

mc regt an, auch mal etwas selbst zu bauen. Denn mc präsentiert Applikationen vom einfachen Interface bis zum kompletten Selbstbausystem.

mc hat auf alle Fragen zur Computertechnik eine Antwort. Mit Hilfe Ihres Computers und eines Telefonmodems können Sie Programme und Literaturstellen direkt bei mc abrufen…

mc kann man ganz einfach kennenlernen. Der nebenstehende Coupon ist dafür bestimmt.

mc
**Die Mikrocomputer-Zeitschrift**

mc

## Kennenlern-Angebot

Ich möchte die mc unverbindlich kennenlernen. Schicken Sie mir die beiden neuesten Ausgaben kostenlos. Informiere ich Sie danach nicht anders, abonniere ich die mc ab _____ zum Jahrespreis von DM 66,– (im Ausland DM 72,–) inkl. Porto.

Name

Beruf

Straße

PLZ/Ort

Datum/Unterschrift

Die Kündigung ist jeweils 8 Wochen vor Ablauf des Abonnements möglich. Wichtig: Nach Erhalt des 2. kostenlosen Heftes kann ich innerhalb von 10 Tagen durch einfache schriftliche Mitteilung an den Verlag von einem Abonnement Abstand nehmen.

Bitte hier Ihre zweite Unterschrift PE2

Bitte ausschneiden und einsenden an:

**Franzis'**
Franzis Verlag, Postfach 37 01 20
8000 München 37

# ProDos für

# Anfänger

## Teil 1: Was ist ProDOS?

**von Ulrich Stiehl**

**Diese Aufsatzserie gibt eine leichtverständliche, aber umfassende Einführung in das neue Diskettenbetriebssystem ProDOS für Applesoft-Programmierer. Da viele Leser bereits unter DOS 3.3 programmiert haben, werden die Unterschiede zwischen ProDOS und DOS 3.3 besonders berücksichtigt. Technische und maschinensprachliche Details werden nur insoweit angesprochen, als sie für die Applesoft-Programmierung unabdingbar sind. Im übrigen wird auf „Apple ProDOS für Aufsteiger", Band 1, verwiesen, das die interne und externe Speicherorganisation von ProDOS in allen Einzelheiten für Assembler-Programmierer behandelt.**

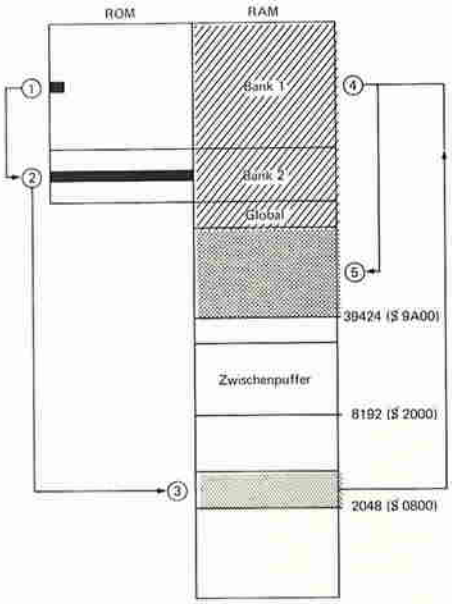
### 1. ProDOS: Pro und Contra

ProDOS ist das im Frühjahr dieses Jahres – in Deutschland anläßlich der Hannover-Messe – vorgestellte neue „professionelle Diskettenbetriebssystem" (Professional Disk Operating System) für den Apple IIc/IIe/II Plus. Ein Diskettenbetriebssystem dient der Übertragung von Daten zwischen externen Diskettenlaufwerken und internem Arbeitsspeicher, z.B. dem Einlesen von Programmen, dem Speichern von Texten usw.
Die Vor- und Nachteile von ProDOS gegenüber DOS 3.3 lassen sich wie folgt charakterisieren:

**Speicherkapazität:** Unter ProDOS kann ein einziger externer Datenträger bis zu 32M (= 32 Megabytes = ca. 32 Millionen Zeichen) umfassen, und eine einzige Datei kann bis zu 16M groß sein. Demgegenüber war DOS 3.3 von vornherein nur für kleinere Datenträger konzipiert gewesen und konnte ohne Modifikationen („Patches") lediglich Disketten mit 140K (= 140 Kilobytes = ca. 140.000 Zeichen) verwalten. Die begrüßenswerte theoretische Speicherkapazität von ProDOS-Datenträgern wird jedoch dadurch entwertet, daß die Firma Apple allem Anschein nach weder willens noch fähig ist, adäquate Datenspeicher anzubieten. Es gibt bekanntlich nur entweder die 140K-Diskettenlaufwerke (Disk II) oder die 5M-Festplattenlaufwerke (Profile). Zu der Disk II paßt ProDOS wie die Faust aufs Auge. Es bringt nichts, ein Moped mit Flugzeugtreibstoff zu tanken. Demgegenüber ist die Profile für viele Zwecke angemessen, doch fehlt die Möglichkeit einer geeigneten Datensicherung. Um den kompletten Inhalt einer Profile auf einer Disk II zu sichern, benötigt man ca. 40 Disketten, und der Kopiervorgang selbst dauert so lange wie das Ausleeren einer Badewanne mit einem Fingerhut. Fazit: Zu ProDOS sollte man Disk-Drives mit einer Speicherkapazität ab 300K aufwärts sowie bei Bedarf Festplattenlaufwerke mit 5M-32M in Verbindung mit adäquaten Datensicherungsgeräten verwenden können. Hierzu wird jedoch von Apple selbst nichts Passendes angeboten, so daß man auf Fremdlieferanten ausweichen muß. Wem indes die normalen 140K-Laufwerke genügen, weil er ohnehin nur begrenzte Daten zu verwalten hat, benötigt in der Regel kein ProDOS.

**Geschwindigkeit:** Die Datenübertragungsrate auf unterster Maschinenebene ist bei ProDOS – ähnlich wie bei dem ebenfalls blockstrukturierten Pascal – mit über 9000 Zeichen/Sekunde erheblich größer als beim alten DOS 3.3 mit über 7000 Zeichen/Sekunde. Diese Werte gelten für Disk-II-Laufwerke. Man muß jedoch berücksichtigen, daß das alte DOS 3.3 1980 erschien. Viele haben vergeblich 4 Jahre lang auf eine verbesserte Version DOS 4.0 gewartet. Da die Firma Apple hier zu lange untätig war, sind von fremden Software-Häusern mittlerweile verbesserte DOS-Varianten herausgebracht worden, z.B. Diversi-DOS usw., die hinsichtlich der Datenübertragungsrate ProDOS kaum nachstehen, ja sogar ProDOS oft überlegen sind.

**Leistungsfähigkeit:** Gegenüber DOS 3.3 sind die Befehle und Dateitypen bei ProDOS ganz beachtlich erweitert worden. Zum einen ist für Assembler-Programmierer eine sehr leistungsfähige Schnittstelle (sog. MLI = Machine Language Interface) geschaffen worden, die kaum Wünsche offenläßt. Zum anderen bieten sich dem




◀ **Diagramm 1:** Stufen des Urladers (vereinfacht):

1 Nach Ctrl-offener Apple-Reset wird durch Power-Up (CALL 64 166) der Controller-Slot gesucht, z. B. Slot 6.
2 Die Controller-Routine (z. B. CALL 50 688 für Slot 6) lädt die Boot-Blocks in den Speicher ab 2048.
3 Das Boot-Block-Programm lädt Systemdatei PRODOS über den Zwischenpuffer ab 8192 in die endgültigen Speicherbereiche in der Sprachkarte und legt die PRODOS Global Page an.
4 PRODOS lädt nunmehr Systemdatei BASIC.SYSTEM über den Zwischenpuffer ab 8192 in den endgültigen Speicherbereich ab 39 424.
5 BASIC.SYSTEM lädt jetzt STARTUP von Diskette, gewöhnlich als Applesoftprogramm nach 2048.

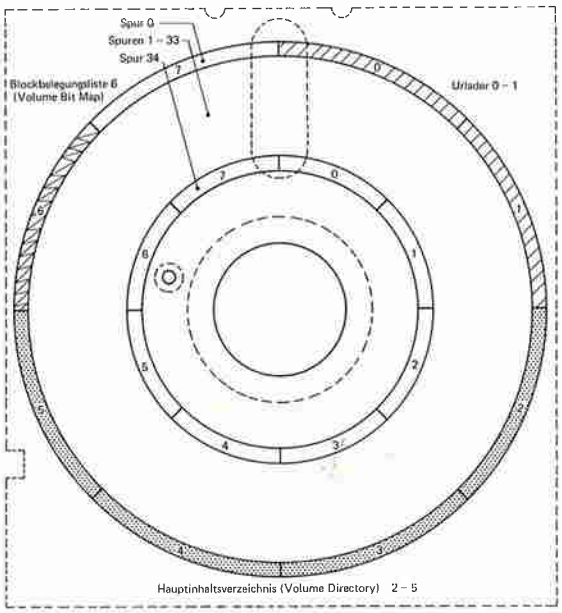



Diagramm 2 35-Spur-Diskette

Applesoft-Programmierer zahlreiche neue Möglichkeiten komfortablerer Programmierung. Stellvertretend sei hier ein Beispiel für DOS-Programmierer genannt: Man kann jetzt auch Textfiles so einladen, als wären es Binärfiles, und man kann darüber hinaus auch beliebige Teile von Binärfiles einlesen, so daß sich speziell für Random-Dateien völlig neue Programmiertechniken eröffnen. Diese erweiterte Leistungsfähigkeit ist in Verbindung mit der größeren Geschwindigkeit der Datenübertragung der entscheidene Vorteil von ProDOS gegenüber dem alten DOS 3.3.

**Sprachkompatibilität:** Das Betriebssystem ProDOS besteht aus zwei Teilen: PRODOS und BASIC.SYSTEM. Der Teil PRODOS – man beachte die Großschreibung gegenüber ProDOS – liegt in den oberen 16K des Apple-Arbeitsspeichers, d.h. in der sog. Language Card (= Sprachkarte), und dient dem eigentlichen Zugriff auf externe Datenträger (Lesen/Schreiben). Der Teil BASIC.SYSTEM liegt in demjenigen Bereich der unteren 48K des Apple-Arbeitsspeichers, wo sich früher das *gesamte* DOS 3.3 befand, und dient der Interpretierung und Abarbeitung der unter Applesoft-BASIC erteilten Befehle wie CATALOG, LOAD, SAVE usw. ProDOS ist damit etwa doppelt so groß wie das alte DOS 3.3. Da durch PRODOS die eigentlich zum Einladen anderer Programmiersprachen gedachte Sprachkarte bereits belegt ist, läuft das BASIC.SYSTEM nur unter Applesoft- und nicht mehr unter Integer-BASIC. Dies ist kein ernsthafter Nachteil, da Integer-BASIC ohnehin nur für Spielprogramme u.ä. verwendet wurde. Erheblich nachteiliger ist jedoch, daß das BASIC.SYSTEM nur für das interpretierende Applesoft konzipiert wurde, d.h. compilierte Applesoft-Programme sind unter ProDOS aus verschiedenen Gründen, die in „Apple ProDOS für Aufsteiger, Band 1" geschildert wurden, grundsätzlich nicht mehr lauffähig. Der Vollständigkeit halber sei hier erwähnt, daß auch ein maschinennaher Diskettenzugriff mittels RDKEY, GETLN, COUT usw. unter ProDOS nicht mehr funktioniert. Daraus folgt, daß compilierte Applesoft- sowie Assembler-Programme nicht auf ProDOS konvertierbar sind. Lediglich lupenreine interpretierende Applesoft-Programme ohne Peeks und Pokes lassen sich ohne großen Aufwand auf ProDOS umstellen.

Aufgrund des obigen Pro und Contra können folgende allgemeine Empfehlungen ausgesprochen werden:

a) Professionelle Assembler-Programmierer, die auf Applesoft verzichten können, sollten ihre neuen Programme unter ProDOS schreiben, auch wenn nur 140K-Laufwerke benutzt werden, da man dann mit dem in der Language Card befindlichen PRODOS, d.h. also ohne das BASIC.SYSTEM, auskommen kann, wodurch kein wertvoller Speicherraum verschwendet wird. Das MLI (= Machine Language Interface) bietet nämlich dem reinen Assembler-Programmierer erheblich mehr Möglichkeiten als die relativ kümmerliche RWTS-Schnittstelle unter DOS 3.3.

b) Applesoft-Programmierer, denen die alten 140K-Laufwerke ausreichen, sollten beim alten DOS 3.3 – in einer „frisierten" Version wie z.B. Diversi-DOS – bleiben, zumal sie dann ihre alten Programme weiterhin verwenden können.

c) Applesoft-Programmierer, die größere Laufwerke, z.B. die in diesem Heft beschriebenen 3,5-Zoll-Sony-Drives, verwenden müssen, sollten auf ProDOS umsteigen, auch wenn sie neben den dann zum Tragen kommenden beachtlichen

Vorteilen von ProDOS einige Nachteile wie etwa die Nicht-Compilierbarkeit ihrer Applesoft-Programme sowie den geringeren freien Arbeitsspeicher in Kauf nehmen müssen.

Das Für und Wider zu ProDOS möchte ich noch mit einigen persönlichen Bemerkungen abschließen. Die Urversion von DOS 3.3, nämlich DOS 3, erschien bereits Mitte 1978. Danach wurden noch weitere Versionen – DOS 3.1, DOS 3.2 und DOS 3.2.1 – herausgebracht, bis schließlich im August 1980 die noch heute gültige Version DOS 3.3 veröffentlicht wurde. Danach war fast 4 Jahre lang „Sendepause". In der Zwischenzeit waren bereits zahlreiche applefremde DOS-Varianten wie David-DOS, Diversi-DOS usw. erschienen, die unter möglichst weitgehender Wahrung der alten Befehlsstruktur das betagte DOS 3.3 bis zu etwa fünfmal schneller machten. Irgendwann vor gut einem Jahr muß dann die Firma Apple in eine Art Torschlußpanik geraten sein. Ein verbessertes DOS 3.3 zu entwickeln war jetzt nicht mehr möglich, da ja bereits verbesserte applefremde DOS-Varianten vorlagen, die Apple selbst wohl kaum mehr hätte übertreffen können. So entschloß man sich dann, das bereits für den Apple III vorliegende SOS-Betriebssystem (Sophisticated Operating System), das ja in 6502-Maschinensprache geschrieben war, kurzerhand für den Apple II umzuschreiben, womit ProDOS entstand. Besser wäre es gewesen, wenn man etwa 1982 oder spätestens 1983 entweder ein DOS 4.0 oder das jetzige ProDOS herausgebracht hätte. Inzwischen liegen viele unter DOS 3.3 lauffähige, ausgereifte Anwender-Programme vor, die wohl nur noch vereinzelt auf ProDOS umgestellt werden dürften. Damit steht dann der Anwender vor dem Problem, sowohl DOS- wie auch ProDOS-Disketten, die miteinander nicht kompatibel sind, verwalten zu müssen, was zu unnötigem Konvertierungsaufwand führt. Nehmen wir als Beispiel das Problem des Applesoft-Compilers. Die Firma Apple hatte es selbst in der Hand, das BASIC.SYSTEM so zu konzipieren, daß die alten Compiler (wie etwa der TASC von der Firma Microsoft, der selbst völlig DOS-neutral ist) auch unter ProDOS lauffähig bleiben. Statt dessen ließ man sich skurrile Dinge wie Pseudo-Trace usw. einfallen, die jeden Compiler aufs Kreuz legen. Wo bleibt dann der Geschwindigkeitsvorteil von ProDOS? Tatsache ist, daß ein unter Diversi-DOS compiliertes Applesoft-Programm spürbar schneller als ein unter ProDOS nicht-compiliertes Applesoft-Programm ist. Und wer schreibt jetzt noch einen Applesoft-Compiler für das BASIC.SYSTEM? Wenn die Firma Apple 4 Jahre brauchte, um eine abgespeckte Version des SOS-Betriebssystems als ProDOS herauszubringen, dann würde sie bei gleicher Effizienz einen eigenen Applesoft-Compiler erst dann herausbringen, wenn der Apple II bereits im Technik-Museum eingemottet ist.

## 2. Monitor, Applesoft und ProDOS

Bei Apple lassen sich drei Arten von Systemprogrammen unterscheiden:

1. Monitor (ROM)
2. Applesoft (ROM)
3. ProDOS (RAM)

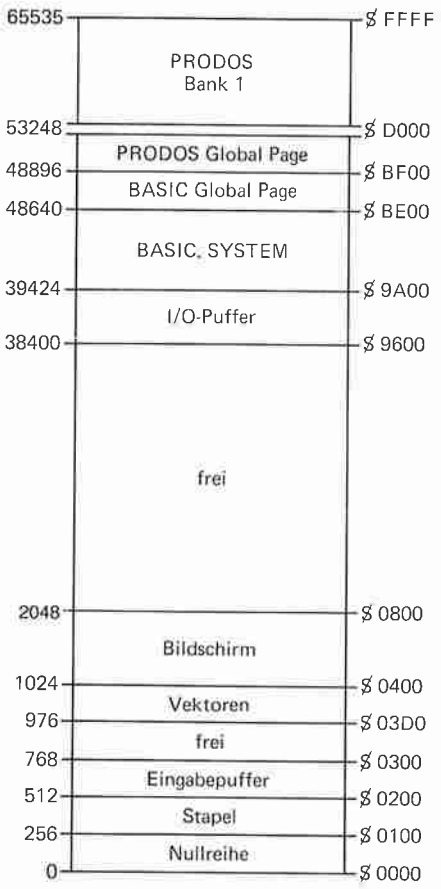
Der **Monitor** ist eine Sammlung von Maschinenroutinen, die u.a. die Bildschirmdarstellung und die Tastaturabfrage übernehmen. Der Monitor befindet sich im ROM (= Read Only Memory = Nur-Lese-Speicher = Festwertspeicher) und ist damit nach dem Einschalten des Apple sofort aktiv. Gäbe es kein Monitorprogramm, dann wäre der Apple nach dem Einschalten „tot", d.h. die Tastatur würde nicht reagieren und am Bildschirm würde man nichts oder nur einen „Buchstabensalat" sehen.

Das **Applesoft** ist eine Sammlung von Maschinenroutinen, die der Interpretierung (= Deutung) und Ausführung von Befehlen der BASIC-Programmiersprache dienen. Applesoft befindet sich ebenfalls im ROM und ist deshalb nach dem Einschalten des Apple wie der Monitor sofort aktiv.

Das **ProDOS** befindet sich nicht im ROM, sondern muß erst von der Diskette in das RAM (= Random Access Memory = Lese-Schreib-Speicher) eingelesen werden. Dieses Einlesen des Betriebssystems wird als **Booten** bezeichnet. Das Bootprogramm oder der sog. Urlader besteht aus mehreren Stufen oder Teilprogrammen.

## 3. Bootvorgang

Nach dem Einschalten des Apple wird der *Urlader Stufe 1* im Monitor (sog. Power-Up-Routine) aktiviert, dessen Zweck darin besteht, herauszufinden, an welchem der bis zu 7 Slots oder Steckplätze eine Controller-Karte für ein Diskettenlaufwerk an-




geschlosssen ist. Wird ein Controller entdeckt, dann wird das Programm auf der Controller-Karte aktiviert. Dies ist der *Urlader Stufe 2*, welcher zunächst den Sektor 0 von Spur 0 der ProDOS-Diskette einliest. Dieses Teilprogramm lädt dann Block 0 und Block 1 von der ProDOS-Diskette, welche zusammen den *Urlader Stufe 3* darstellen, der seinerseits das Betriebssystem namens PRODOS nunmehr vollständig in das RAM überträgt. Nachdem das Betriebssystem PRODOS in der Language Card (= Sprachkarte) installiert worden ist, wird noch das sog. BASIC.SYSTEM eingeladen, das dann seinerseits das Hello-Programm namens STARTUP einliest, womit der Bootvorgang abgeschlossen wird.

*Hinweis:* Unter Hello-Programm versteht man ein Programm, das im Anschluß an die Installierung des Betriebssystems automatisch gestartet wird. Bei ProDOS muß das Hello-Programm STARTUP heißen, während bei DOS 3.3 ein beliebiger Dateiname gewählt werden konnte. „Booten" kommt von englisch „to bootstrap" und bedeutet „an der Stiefelschlaufe ziehen"

oder im übertragenen Sinne „sich wie Münchhausen am eigenen Schopf herausziehen".

Um ProDOS zu starten, gibt es eine Reihe von Möglichkeiten, wobei jeweils vorausgesetzt wird, daß sich eine bootfähige ProDOS-Diskette in einem Laufwerk befindet. Für (1) und (2) gilt, daß sich die ProDOS-Diskette in Laufwerk 1 befinden muß. Demgegenüber kann sich bei (3) die ProDOS-Diskette auch in Laufwerk 2 befinden oder der externe Datenträger kann eine RAM-Disk oder ein Festplattenlaufwerk sein. Beim Apple IIc kann ProDOS auch von externen Laufwerken gebootet werden.

1. Wenn der Apple noch nicht eingeschaltet ist (echter Kaltstart):

1.1. Einschalten des Stromes
Bewirkt Kaltstart über Urlader Stufe 1; funktioniert immer.

2. Wenn der Apple bereits eingeschaltet ist („lauwarmer" Kaltstart):

2.1. Ctrl-Offener Apfel-Reset
beim Apple IIc oder IIe. Dies entspricht dem Kaltstart über Urlader Stufe 1; funktioniert immer.

2.2. CALL 64166
Dieser Applesoft-Befehl ruft den Urlader Stufe 1 im Monitor auf; funktioniert nur, wenn Applesoft aktiv ist. CALL 64166 ermittelt automatisch den Controller, der sich im Slot mit der höchsten Nummer – meist Slot 6 – befindet.

2.3. PR#6 oder CALL 50688
Die Applesoft-Befehle PR#6 oder CALL 50688 rufen unter Umgehung des Urladers Stufe 1 direkt den Urlader Stufe 2 im Controller in Slot 6 auf. Wenn man sich gerade im Monitor-Modus befindet – erkennbar an dem *-Prompt –, so kann man statt PR#6 auch
6 Ctrl-P
sagen; funktioniert nur, wenn Slot 6 mit einem Controller belegt ist.

3. Wenn sich ProDOS noch im Speicher befindet („warmer" Kaltstart):

3.1. –PRODOS
Dieser sog. Strichbefehl in Verbindung mit dem Dateinamen PRODOS startet ProDOS neu; funktioniert nur, wenn sich sowohl das PRODOS als auch das BASIC.-SYSTEM noch intakt im Speicher befinden. Zweck: Booten von einem nicht über Kaltstart bootfähigen Datenspeicher wie Profile, 3,5-Zoll-Laufwerk usw.

3.2. Reboot-Programm
In der Language Card wird von ProDOS ein sog. Reboot-Programm installiert, das von Applesoft allerdings nur über ein kleines Maschinenprogramm aufgerufen werden kann. Der Vollständigkeit halber sei bereits an dieser Stelle gezeigt, wie man dies in Applesoft machen kann:

POKE 768, 32: POKE 769, 0: POKE 770, 191: POKE 771, 101: POKE 772, 6: POKE 773, 3: POKE 774,4: CALL 768

Nach den Pokes und nach CALL 768 erscheint zunächst die Meldung
ENTER PREFIX (PRESS „RETURN" TO ACCEPT)
worauf man das gewünschte Präfix eingeben muß. Danach erscheint als zweite Meldung
ENTER PATHNAME OF NEXT APPLICATION
worauf man den gewünschten Pfadnamen eingeben muß.
Das Reboot-Programm setzt voraus, daß sich PRODOS noch intakt in der Language Card befindet, während das BASIC.SYSTEM nicht benötigt wird.

*Hinweis:* Vom echten Kaltstart spricht man, wenn ein Systemprogramm und Systemwerte völlig neu initialisiert werden. Bei geschützten Anwenderprogrammen oder beim Wechsel von einem Prozessor zum anderen (z.B. vom Z80 zum 6502) geht dies oft nur über Ausschalten-Einschalten. In allen anderen Fällen sind unterschiedlich „warme" Kaltstartformen möglich. Vom echten Warmstart spricht man, wenn man zwangsweise, jedoch in der Regel ohne Zerstörung von bereits initialisierten Werten, in einen übergeordneten Befehlsmodus gelangt, z.B. wenn man ein Applesoft-Programm mit Ctrl-Reset oder Ctrl-C unterbricht.

## 4. Externe Speicherorganisation

Unter ProDOS können als externe Datenspeicher verwendet werden:

a) Disk-II-Laufwerke (140 Kilobytes)
b) 64K-Karte als RAM-Disk beim IIc/IIe
c) Profile-Festplattenlaufwerk (5 Megabytes)
d) Größere Laufwerke (3,5 Zoll Sony, 5,25 Zoll Teac usw. mit mehr als 300K)
e) Größere RAM-Karten als RAM-Disks (mit mehr als 128K)
f) Größere Festplattenlaufwerke (Corvus usw. mit mehr als 20M)

Für die kleineren externen Datenspeicher sind von der Firma Apple eigene Driver-Programme entwickelt worden (Disk-Driver = Treiber- oder Steuerprogramm für Controller-Karten). Für die leistungsfähigeren und größeren Datenspeicher befinden sich Driver-Programme von Fremdfirmen in Vorbereitung. Für manche Laufwerke wie etwa die Sony-Drives sind nicht einmal gesonderte Driver erforderlich. Hier genügt ein entsprechendes Formatierungsprogamm wie unser PRODOS.INIT, das in Heft 1/84 von „Peeker" besprochen wurde.

Nachfolgend beschränken wir uns zunächst auf die normalen Apple-Disketten für Disk-II-Laufwerke. Für diese 5,25-Zoll-Disketten gilt:

– Jede Diskette hat 35 konzentrische Spuren, numeriert von 0-34.
– Jede Spur ist in 8 Blocks unterteilt.
– Jeder Block umfaßt 512 Bytes oder Zeichen

Eine Diskette enthält damit 35 Spuren mal 8 Blocks = insgesamt 280 Blocks, die von 0-279 durchnumeriert sind und von denen die ersten 7 Blocks auf Spur 0 einen festen Inhalt haben:

Blocks 0 bis 1 enthalten den Urlader Stufe 2
Blocks 2 bis 5 enthalten das sog. Volume Directory oder Hauptinhaltsverzeichnis
Block 6 enthält die sog. Volume Bit Map, in der über die belegten Blocks „Buch geführt" wird.

Die restlichen 273 Blocks sind theoretisch für Programme und Daten frei. Eine unter Applesoft bootfähige Diskette muß indessen neben den 7 Blocks für Urlader usw. noch mindestens 3 weitere Dateien enthalten, nämlich

a) Datei PRODOS (31 Blocks), die vom Urlader Stufe 2 geladen wird
b) Datei BASIC.SYSTEM (21 Blocks), die vom PRODOS geladen wird
c) Datei STARTUP (mindestens 2 Blocks), die vom BASIC.SYSTEM geladen wird.

Damit gilt folgende Raumaufteilung:

7 Blocks für Urlader usw.
31 Blocks für PRODOS
21 Blocks für BASIC.SYSTEM
2 Blocks für STARTUP

# Hüthig-FACHBUCH-TIP

„Apple Assembler" wendet sich an alle, die bereits Anfängerkenntnisse der 6502-Programmierung haben und nunmehr ein Nachschlagewerk für ihren Apple II Plus/II e/II c suchen, in dem alle wichtigen ROM-Routinen sowie eine Vielzahl sonstiger Hilfsprogramme in einer systematischen Form zusammengestellt werden. Insgesamt umfaßt dieses Buch über 40 Utilities, darunter mehrere völlig neuartige Programme wie Double-Lores, Double-Hires, Screen-Format u. a.

Der erste Teil enthält ein Repetitorium der wichtigsten Befehle, Adressierungsarten und sonstigen Besonderheiten des 6502 sowie Angaben zu den apple-spezifischen Zahlenformaten (Integer- und Fließkommazahlen).
Im zweiten Teil werden neben einer Kurzwiederholung der Monitor-Befehle alle Routinen und Adressen des Monitors zusammengestellt, die für Assemblerprogrammierer von Nutzen sein können. Darüber hinaus findet der Leser Unterroutinen für Vorwärts- und Rückwärts-Moven, hexadezimale Addition/Subtraktion/Multiplikation/Division, Binär-, Hex-und ASCII-Umwandlung, Dumpen/Disassemblieren, Aufwärts-Scrollen, Reset u. a.
Der dritte Teil befaßt sich mit der Speicherverwaltung der Language-Card und der II e-64K-Karte und enthält Test- und Move-Programme zum Verschieben von Daten in die und aus der Language Card sowie der 64K-Karte, wobei besonders ausführlich auf die Softswitches eingegangen wird.

Der vierte Teil ist dem Applesoft-ROM gewidmet und beschreibt die interne Struktur von Applesoft-Programmen, die Methoden der Parameterübergabe mittels CALL, USR, &, PEEK und POKE und listet dann eine große Anzahl nützlicher Interpreter-Adressen. Bei den Utility-Programm liegt das Schwergewicht auf Fließkommamathematik einschließlich Print Using.

Der letzte Teil behandelt den Text- und Grafikspeicher. Neben einem professionellen Maskengeneratorprogramm findet der Leser hier auch erstmals Routinen zur Double-Lores- und Double-Hires-Grafik des Apple II e.

## Apple Assembler - Tips und Tricks -

**von U. Stiehl**

232 S., 40 Programm-Listings, 3 Abb., kart., DM 34,—
ISBN 3-7785-1047-9

Begleitdiskette zum Buch DM 28,—
ISBN 3-7785-1048-7

BESTELLCOUPON

Bitte ausfüllen und an Hüthig Vertriebsservice, Postfach 102869 · 6900 Heidelberg schicken.

Buchtitel

Name

Straße

Ort

Unterschrift

Dies sind zusammen 61 Blocks. Damit verbleiben nur noch 280 minus 61 = 219 Blocks für sonstige Programme und Daten. Dies sind 219 mal 512 = 112.128 Bytes = 109.5K Nettospeicherkapazität, also nicht allzuviel für ein Betriebssystem, das 32M brutto verwalten könnte.

Unter DOS 3.3 sieht demgegenüber die Rechnung folgendermaßen aus: Jede der 35 Spuren wird nicht in 8 Blocks zu je 512 Bytes, sondern in 16 Sektoren zu je 256 Bytes unterteilt. Damit umfaßt eine DOS-Diskette 35 mal 16 = 560 Sektoren. Das DOS-Betriebssystem belegt die Spuren 0 bis 2, und das Inhaltsverzeichnis liegt auf Spur 17. Damit sind von vornherein 4 mal 16 = 64 Sektoren belegt. Hinzu kommt noch ein mindestens 2 Sektoren umfassendes Hello-Programm, mithin insgesamt 66 Sektoren. Damit verbleiben 560 minus 66 = 494 Sektoren für sonstige Programme und Daten. Dies sind 494 mal 256 = 126.464 Bytes = 123.5K (gegenüber 109.5K bei ProDOS-Disketten). Daraus folgt, daß auf einer DOS-Diskette 123.5K minus 109.5K = 14K mehr Platz ist.

*Hinweis:* 1 Byte = 1 Zeichen (z.B. Buchstabe, Ziffer, Code). 1K = 1 Kilobyte = 1024 Bytes. 1M = 1 Megabyte = 1024K. Merke ferner als applespezifisch: 1 Block = 512 Bytes. 1 Sektor = 256 Bytes.

## 5. Interne Speicherorganisation

Applesoft-Programmierer müssen zumindest einen groben Überblick über die interne Speicherorganisation von ProDOS haben. Bekanntlich kann der Mikroprozessor des Apple II 64K oder exakt 65536 Speicherstellen adressieren bzw. verwalten. Diese Speicherstellen sind von 0-65535 durchnumeriert. Unter ProDOS und Applesoft gilt dann folgende Einteilung (Adressen in dezimal):

**00000-00255:** Nullseite mit wichtigen Systemwerten; normalerweise für Applesoft tabu

**00256-00511:** Prozessor-Stapelspeicher; normalerweise für Applesoft tabu

**00512-00767:** Tastatur-Eingabepuffer; wird außerdem vom BASIC.SYSTEM während des Diskettenzugriffs zur Zwischenspeicherung des Dateinamens usw. benutzt. Deshalb darf die sog. GETLN-Tastatur-Eingaberoutine nicht aufgerufen werden, während ein Textfile geöffnet ist.

**00768-00975:** Frei für kurze Maschinenprogramme

**00976-01023:** Vektoren für ProDOS-Warmstart, Reset, Interrupt usw. (gegenüber DOS 3.3 nur noch teilweise benutzt)

**01024-02047:** 40-Z/Z-Bildschirmspeicher

**02048-38399:** Frei für Applesoft-Programm, Hires-Grafik usw.

**38400-39423:** HIMEM (Highest Memory = Obergrenze des freien Speichers) und zugleich Beginn des ersten BASIC.SYSTEM-Puffers für Input/Output. Wenn mehr als 1 Datei geöffnet ist, verschiebt sich HIMEM vorübergehend um je 1024 Bytes pro Puffer nach unten.

**39424-48639:** BASIC.SYSTEM mit internen Puffern

**48640-48895:** Sog. BASIC.SYSTEM Global Page mit Systemwerten

**48896-49151:** Sog. PRODOS Global Page mit Systemwerten

**49152-53247:** ROM-Bereich für die Slots 0-7

**53248-65535:** ROM-Bereich des Applesoft-Interpreters und des Monitors

**53248-65535:** RAM-Bereich der Sprachkarte Bank 1 mit dem eigentlichen PRODOS

**53248-57343:** RAM-Bereich der Sprachkarte Bank 2 mit Reboot-Programm

Damit steht dem Applesoft-Programmierer wie unter DOS 3.3 erstens der Bereich 2048-38399 für das eigentliche Applesoft-Programm sowie der Bereich 768-975 für kurze Maschinenprogramme zur Verfügung. Der unter DOS 3.3 früher unbenutzte 16K-Bereich der Sprachkarte ist nunmehr durch PRODOS sowie das Reboot-Programm belegt. Damals konnte man z.B. das DOS 3.3 komplett in die Sprachkarte schieben und hatte dann die unteren 48K fast völlig zur freien Verfügung, so daß sehr große Applesoft-Programme erstellt werden konnten. Diese Möglichkeit ist unter ProDOS entfallen.

Unter DOS 3.3 gab es den bei ProDOS nicht mehr existierenden Befehl MAXFILES, mit dessen Hilfe man eine gewünschte Anzahl von Puffern reservieren konnte. Darüber hinaus konnte man HIMEM herabsetzen und größere Maschinenprogramme zwischen HIMEM und dem Beginn des untersten DOS-Puffers einladen. Auf dieses Problem sowie die damit zusammenhängende Garbage Collection bei dynamischen Puffern wird in einem der nachfolgenden Aufsätze dieser ProDOS-Serie eingegangen.

*Anmerkung zu den Diagrammen:*

**Diagramm 1** zeigt in vereinfachter Form die verschiedenen Stufen des Urladers. Hierzu ist zu präzisieren, daß PRODOS und BASIC.SYSTEM in Wahrheit nicht direkt in ihren endgültigen RAM-Bereich, sondern zunächst in den Speicher ab $2000 eingelesen und dann von dort an ihren definitiven Platz verschoben werden. PRODOS wird nur dann komplett eingeladen bzw. verschoben, wenn ein Apple II Plus, IIe oder IIc vorliegt. Für Nachbauten sind diverse Patches erforderlich (siehe Heft 1/84 von „Peeker"). Das BASIC.SYSTEM wird nur dann eingelesen, wenn der Apple II Plus mit einer Sprachkarte ausgestattet ist. Beim IIe und IIc ist dies automatisch der Fall, während Apple II Plus Geräte mit 48K entsprechend aufgerüstet werden müssen.

PRODOS ließe sich in einen 48K-Apple einlesen und würde dann etwa denjenigen Platz einnehmen, den sonst bei einem 64K-Apple das BASIC.SYSTEM einnimmt. Assembler-Programmierer, die das 48K-PRODOS mit dem 64K-PRODOS vergleichen wollen, aber einen 64K-Apple besitzen, können PRODOS mit folgenden Patches „überlisten":

```
BLOAD PRODOS, A$2000, TSYS
CALL -151
23B9: D0
23EA: F0
242B: 38
239A: D0
2000G
```

Danach wird PRODOS in den hinteren Bereich der 48K und nicht in die Language Card verschoben.

**Diagramm 2** zeigt schematisch die Organisation einer 35-Spur-Diskette für Disk-II-Laufwerke.

**Diagramm 3** veranschaulicht die interne Speicherorganisation unter ProDOS bei einem 64K-Apple. Bei einem Apple IIc bzw. einem Apple IIe mit 64K-Karte käme noch die 64K-RAM-Disk hinzu, die wahlweise mit Slot 3, Drive 2 oder mit dem Volume-Namen /RAM angesprochen wird.

# APPLE
# MASCHINEN SPRACHE

## Der Einstieg über BASIC!

**Die Idee:** Maschinenbefehle des Mikroprozessors 6502 über die BASIC-Befehle POKE, CALL, PEEK im Apple abspeichern, ablaufen lassen und Ergebnisse in das BASIC-Programm zurücklesen. Grundkenntnisse in Basic sind erforderlich. Anschauliche Darstellung der 6502-Funktionen, der Speicher im Apple und der Wirkungen von Maschinenbefehlen – verbunden mit Experimentierprogrammen zur Grafik, Akustik, Arithmetik, Textdarstellung, Verbindung von Bild und Ton sowie zum Aufruf der zahlreichen, bereits vorhandenen Maschinenprogramme im Apple.

**Der Höhepunkt:** Eine schrittweise Einführung in die professionelle Entwicklung von Maschinenprogrammen mit APPLE SYSTEM-MONITOR und APPLE MINI ASSEMBLER.

**Von D. Inman/K. Inman, 224 Seiten, Softcover, DM 49,–**

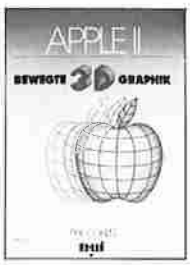
te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

# Weiterführende Literatur...

**APPLE II – Anwenderhandbuch**
(L. Poole)
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was herstellerseitig an Literatur angeboten wird.
416 Seiten, Softcover, DM 56,–

**LOGO – Jeder kann programmieren**
(Daniel Watt)
Buch des Jahres in den USA. Für die Computer C64, ATARI, APPLE II, IBM-PC und TI-99.
Hochwertiges Textbuch für Logo-Kurse für zu Hause und im Lehrbereich.
A4, DM 59,– (4. Quartal 84)

**APPLE II PASCAL – Eine praktische Anleitung**
(A. Luehrmann, H. Peckham)
Unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen und Zugang zu einem Apple-Computer haben.
544 Seiten, Softcover, DM 59,–




**APPLE II – Bewegte 3D-Graphik**
(Phil Cohen)
Selbstentworfene Graphiken und Diagramme – animiert oder als Standbilder – eben oder räumlich: alle erforderlichen BASIC-Programme mit Erklärung finden Sie in diesem Buch.
ca. 190 Seiten, Softcover, DM 49,–
(4. Quartal 84)

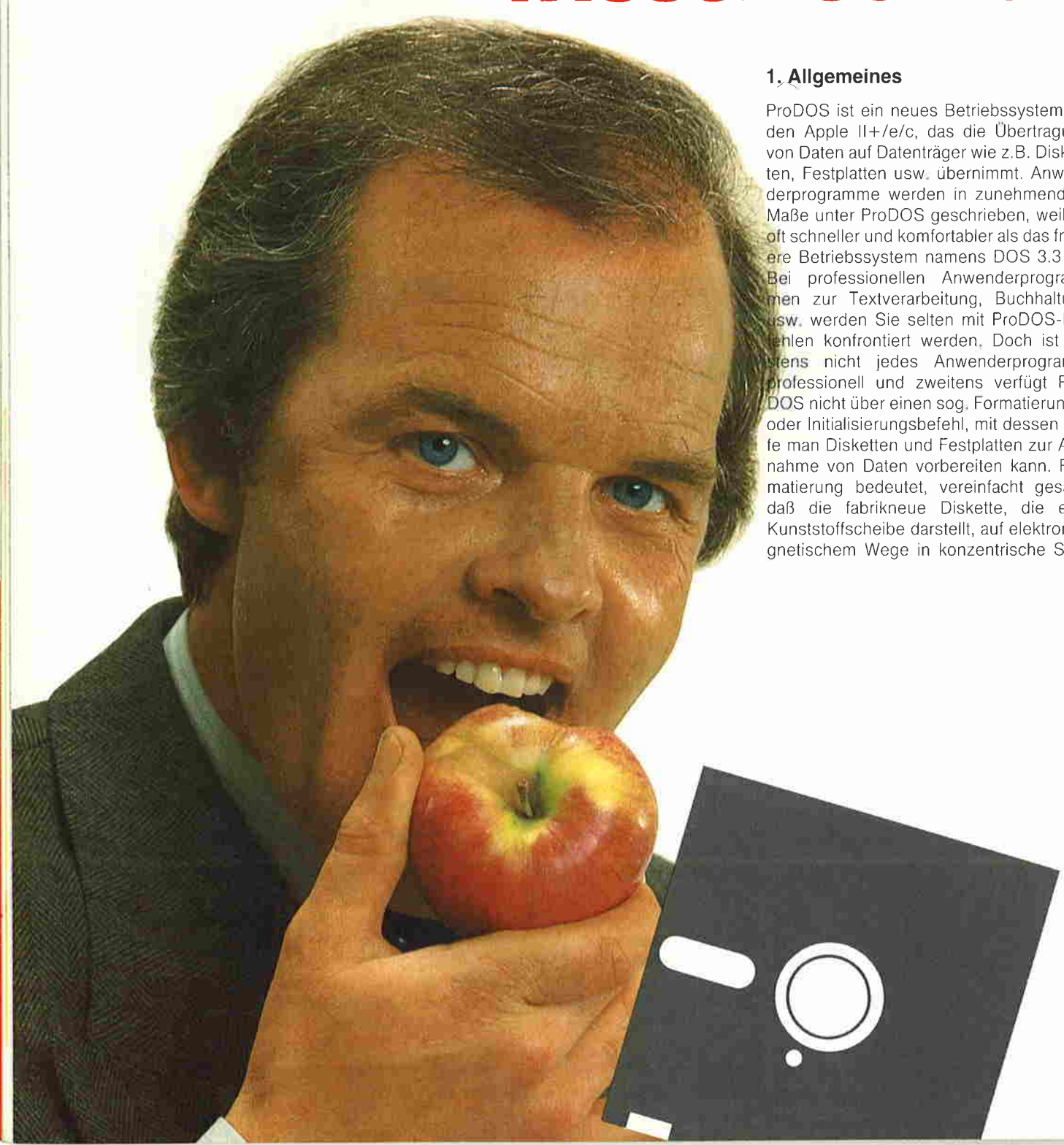
**Computer für Kinder**
(Sally Greenwood Larson)
Ein Buch für Kinder, ihre Lehrer und Eltern.
„Computer für Kinder" richtet sich an Kinder im Alter von 8 bis 13 Jahren, für deren Interesse an Computern dieses Buch bewußt geschrieben wurde.
Unterhaltsam und leicht verständlich.
A4 quer, Fadenheftung, DM 29,80

**Macintosh Programmier-Handbuch**
(David Lien)
Macintosh – Pionier des graphischen Computerdialogs – benutzt MICROSOFT BASIC. D. Lien's Buch ist eine souveräne, zuverlässige Einführung und Referenz zu Macintosh's Basic.
450 Seiten, DM 59,– (4. Quartal 84)

im 4104

**Noch im Programm: 6502 – Programmieren in Assembler DM 59,–**
**VisiCalc, 50 Programme auf Diskette, DM 79,–**

# Was Geschäftsleute vom ProDOS-FILER wissen sollten



## 1. Allgemeines

ProDOS ist ein neues Betriebssystem für den Apple II+/e/c, das die Übertragung von Daten auf Datenträger wie z.B. Disketten, Festplatten usw. übernimmt. Anwenderprogramme werden in zunehmendem Maße unter ProDOS geschrieben, weil es oft schneller und komfortabler als das frühere Betriebssystem namens DOS 3.3 ist. Bei professionellen Anwenderprogrammen zur Textverarbeitung, Buchhaltung usw. werden Sie selten mit ProDOS-Befehlen konfrontiert werden. Doch ist erstens nicht jedes Anwenderprogramm professionell und zweitens verfügt ProDOS nicht über einen sog. Formatierungs- oder Initialisierungsbefehl, mit dessen Hilfe man Disketten und Festplatten zur Aufnahme von Daten vorbereiten kann. Formatierung bedeutet, vereinfacht gesagt, daß die fabrikneue Diskette, die eine Kunststoffscheibe darstellt, auf elektromagnetischem Wege in konzentrische Spu-

ren eingeteilt wird. Eine normale Apple-Diskette besteht nach der Formatierung aus 35 Spuren, wobei jede Spur etwa 4000 Zeichen (Buchstaben oder Ziffern) speichern kann. ProDOS unterteilt diese Spuren in noch kleinere Einheiten, die sog. Blocks. Jeder Block umfaßt etwa 500 Zeichen. Wenn auf eine Spur 4000 Zeichen passen, dann umfaßt eine Spur 4000 : 500 = 8 Blocks. Und wenn auf eine Diskette 35 Spuren passen, dann umfaßt eine Diskette 35 · 8 = 280 Blocks. Insgesamt passen auf eine solche Diskette etwa 140.000 Zeichen.

Jede Diskette hat – ähnlich wie ein Buch – ein Inhaltsverzeichnis. Während im Buchinhaltsverzeichnis die Kapitel aufgeführt sind, sind im Disketteninhaltsverzeichnis die Dateien genannt. Eine Datei kann ein Brief, eine Rechnung, eine Adreßkartei oder auch ein Programm sein. Die Namen der Dateien müssen in Großbuchstaben geschrieben werden und dürfen höchstens 15 Buchstaben lang sein, z.B. WERBEBRIEF, KUNDENSTAMM, TEXTPROGRAMM usw. Bei ProDOS haben jedoch nicht nur die Dateien einen Namen, sondern auch die Diskette selbst. Man muß also zwischen Diskettennamen und Dateinamen unterscheiden. Ähnlich wie man ein Buch in mehrere Bände aufteilen kann, läßt sich auch eine ProDOS-Diskette in mehrere getrennte Teile gliedern, z.B. folgendermaßen:

1. DISKETTE.NR.10 (Diskettenname)
1.1. BRIEFE (Teilname)
1.1.1. PRIVAT (Dateiname)
1.1.2. DIENSTLICH (Dateiname)
1.2. PROGRAMME (Teilname)
1.2.1. BUCHHALTUNG (Dateiname)
1.2.2. TEXTPROGRAMM (Dateiname)

Bei diesem Beispiel heißt die Diskette „DISKETTE.NR.10". Dies ist der Name des Hauptinhaltsverzeichnisses (Volume Directory). Die Diskette selbst besteht aus zwei Teilen: BRIEFE und PROGRAMME. Dies sind die Namen der Unterinhaltsverzeichnisse (Subdirectory). Wenn eine Datei namens „/DISKETTE.NR.10/BRIEFE/DIENSTLICH" von der Diskette eingeladen wird, so meint man damit die Datei „DIENSTLICH" *im* Unterinhaltsverzeichnis „BRIEFE" *im* Hauptinhaltsverzeichnis „DISKETTE.NR.10". Wie ersichtlich, müssen die Namen des Hauptinhaltsverzeichnisses, des Unterinhaltsverzeichnisses und der eigentlichen Datei durch Schrägstriche abgegrenzt werden, damit ProDOS weiß, welche Inhaltsverzeichnis-*Hierarchie* vorliegt.

## 2. Das Kopierprogramm FILER

Um unabhängig von einem Anwenderprogramm Disketten formatieren oder Dateien kopieren zu können, kann man das Programm FILER benutzen, das auf der ProDOS-Systemdiskette namens USERS.-DISK enthalten ist, welche in Verbindung mit dem „ProDOS User's Manual" über Apple-Händler erhältlich ist. Der FILER wird gestartet, indem man die USERS.-DISK in das Laufwerk 1 einlegt, den Apple einschaltet und dann aus der Bildschirmübersicht „F„ für FILER auswählt. Leider sind sowohl der FILER wie auch das „ProDOS User's Manual" in englischer Sprache abgefaßt. Eine deutsche Übersetzung des Benutzerhandbuches soll im ersten Quartal 1985 erscheinen. Der FILER selbst sowie die ProDOS-Befehle dürften wohl nicht eingedeutscht werden, so daß wir nachfolgend alle englischen Bildschirmübersichten des FILERs abbilden und kurz erläutern wollen.

Bild 1: Hauptübersicht des FILERs

```
F - Dateibefehle
V - Diskettenbefehle
D - Ersatzwerte
Q - Ende

****************************************
*                                      *
*   APPLE'S PRODOS SYSTEM UTILITIES    *
*                                      *
*        FILER  VERSION 1.0.1          *
*                                      *
*  COPYRIGHT APPLE COMPUTER,  1983-84  *
*                                      *
****************************************

? - TUTOR
F - FILE COMMANDS
V - VOLUME COMMANDS
D - CONFIGURATION DEFAULTS
Q - QUIT
```

Bild 2: Übersicht über Dateibefehle

```
L - Inhaltsverzeichnis ansehen
C - Datei kopieren
D - Datei löschen
K - Dateien vergleichen
A - Schreibschutz umkehren
R - Dateiname ändern
M - Inhaltsverzeichnis anlegen
P - Ersatz-Teilname festlegen

****************************************
*                                      *
*             FILE COMMANDS            *
*                                      *
****************************************

? - TUTOR
L - LIST PRODOS DIRECTORY
C - COPY FILES
D - DELETE FILES
K - COMPARE FILES
A - ALTER WRITE PROTECTION
R - RENAME FILES
M - MAKE DIRECTORY
P - SET PREFIX
```

Bild 3: Inhaltsverzeichnis ansehen

```
****************************************
*                                      *
*        LIST PRODOS DIRECTORY         *
*                                      *
**PREFIX: /PEEKER/********************
--DIRECTORY--
   PATHNAME: (/PEEKER
                                      )
```

Bild 4: Inhaltsverzeichnis "/PEEKER"

```
DIRECTORY:/PEEKER
 NAME              TYP  BLOCKS  MODIFIED
*PRODOS.EDITOR     BIN     13    1-AUG-84
 BILDSCHIRM        BIN      1   10-OCT-84
 RAMSUCHER         BIN      1   21-JUN-84
 BLOCKS FREE:   258       USED:     22
```

Bild 5: Datei kopieren

```
****************************************
*                                      *
*             COPY FILES               *
*                                      *
**PREFIX: /PEEKER/********************
--COPY--
   PATHNAME: /PEEKER/BILDSCHIRM
    TO PATHNAME: (/STIEHL/BILDSCHIRM
  )
```

Bild 6: Datei löschen

```
****************************************
*                                      *
*            DELETE FILES              *
*                                      *
**PREFIX: /PEEKER/********************
--DELETE--
   PATHNAME: (/PEEKER/RAMSUCHER
                                      )
```

Bild 7: Dateien vergleichen

```
****************************************
*                                      *
*            COMPARE FILES             *
*                                      *
**PREFIX: /PEEKER/********************
--COMPARE--
   PATHNAME: /PEEKER/BILDSCHIRM
    TO PATHNAME: (STIEHL/BILDSCHIRM
  )
```

Bild 8: Schreibschutz setzen/aufheben
(Sternchen = schreibgeschützt)

```
****************************************
*                                      *
*       ALTER WRITE-PROTECTION         *
*                                      *
**PREFIX: /PEEKER/********************
--ALTER WRITE-PROTECTION--
   PATHNAME: (/PEEKER/PRODOS.EDITOR
                                      )
```

Bild 9: Dateiname ändern

```
****************************************
*                                      *
*            RENAME FILES              *
*                                      *
**PREFIX: /PEEKER/********************
--RENAME--
   PATHNAME:  /PEEKER/PRODOS.EDITOR
   NEW PATHNAME: (/PEEKER/BASIC.EDITOR
  )
```

Bild 10: Neues Inhaltsverzeichnis anlegen

```
****************************************
*                                      *
*            MAKE DIRECTORY            *
*                                      *
**PREFIX: /PEEKER/*********************
--MAKE DIRECTORY--
   PATHNAME: (/PEEKER/AUFSAETZE
                                      )
```

Bild 11: Inhaltsverzeichnis anzeigen
(? steht für Ersatz-Diskettenname)

```
****************************************
*                                      *
*         LIST PRODOS DIRECTORY        *
*                                      *
**PREFIX: /PEEKER/*********************
--DIRECTORY--
   PATHNAME: (?
                                      )
```

Bild 12: Ersatz-Teilname festlegen
= Kombination von Namen von Inhaltsverzeichnissen, die dem eigentlichen Dateinamen vorangestellt wird, z.B. "/PEEKER/AUFSAETZE"

```
****************************************
*                                      *
*              SET PREFIX              *
*                                      *
**PREFIX: /PEEKER/*********************
--SET PREFIX--
   NEW PREFIX: (/PEEKER/AUFSAETZE
)
```

Bild 13: Inhaltsverzeichnis nach Änderung des Ersatz-Teilnamens

```
DIRECTORY:/PEEKER/AUFSAETZE
 NAME              TYP  BLOCKS  MODIFIED
0 FILES
 BLOCKS FREE:   257       USED:    23
```

---

Bild 14: Übersicht über Diskettenbefehle
F = Diskette formatieren
C = Diskette kopieren
L = Diskettennamen anzeigen
R = Diskettenname ändern
D = Defekte Blocks ermitteln
B = Freie Blocks ermitteln
K = Disketten vergleichen

```
****************************************
*                                      *
*           VOLUME COMMANDS            *
*                                      *
****************************************
         ? - TUTOR
         F - FORMAT A VOLUME
         C - COPY A VOLUME
         L - LIST VOLUMES
         R - RENAME A VOLUME
         D - DETECT BAD BLOCKS
         B - BLOCK ALLOCATION
         K - COMPARE VOLUMES
```

Bild 15: Diskette formatieren
(Da eine zu formatierende Diskette noch keinen Namen hat, muß man zunächst den Steckplatz und die Laufwerknummer angeben.)

```
****************************************
*                                      *
*            FORMAT A VOLUME           *
*                                      *
****************************************
--FORMAT--
  THE VOLUME IN SLOT:  6
                DRIVE:  2
  NEW VOLUME NAME: (/BLANK00          )
```

Bild 16: Diskette kopieren

```
****************************************
*                                      *
*             COPY A VOLUME            *
*                                      *
****************************************
--COPY--
  THE VOLUME IN SLOT:  6
                DRIVE:  1
   TO VOLUME IN SLOT:  6
                DRIVE:  2
 NEW VOLUME NAME: (/PEEKER           )
```

Bild 17: Diskettennamen anzeigen

```
****************************************
*                                      *
*            LIST VOLUMES              *
*                                      *
****************************************
     SLOT  DRIVE      VOLUME NAME
       3     2       /RAM
       6     1       /PEEKER
       6     2       /STIEHL
```

Bild 18: Diskettenname ändern

```
****************************************
*                                      *
*           RENAME A VOLUME            *
*                                      *
****************************************
--RENAME--
  THE VOLUME IN SLOT:  6
                DRIVE:  1
  NEW VOLUME NAME: (/APPLE.MAGAZIN  )
```

Bild 19: Erneut Diskettennamen anzeigen

```
****************************************
*                                      *
*            LIST VOLUMES              *
*                                      *
****************************************
     SLOT  DRIVE      VOLUME NAME
       3     2       /RAM
       6     1       /APPLE.MAGAZIN
       6     2       /STIEHL
```

Bild 20: Defekte Blocks ermitteln

```
****************************************
*                                      *
*           DETECT BAD BLOCKS          *
*                                      *
****************************************
--DETECT BAD BLOCKS--
  FOR VOLUME IN SLOT:  6
                DRIVE: (1)
```

Bild 21: Freie Blocks ermitteln

```
****************************************
*                                      *
*           BLOCK ALLOCATION           *
*                                      *
****************************************
--BLOCK ALLOCATION--
  FOR VOLUME IN SLOT: (6)
                DRIVE:  1
             23 BLOCKS USED
            257 BLOCKS FREE
            280 BLOCKS TOTAL
```

Bild 22: Disketten vergleichen

```
****************************************
*                                      *
*            COMPARE VOLUMES           *
*                                      *
****************************************
--COMPARE--
  THE VOLUME IN SLOT:  6
                DRIVE:  1
   TO VOLUME IN SLOT:  6
                DRIVE: (2)
```

---

Bild 23: Ersatzwerte festlegen
(M = Bildschirm)

```
****************************************
*                                      *
*            SELECT DEFAULTS           *
*                                      *
****************************************
--SELECT DEFAULTS--
     FOR SOURCE SLOT:  6
                DRIVE:  1
   DESTINATION SLOT:  6
                DRIVE:  2
 SELECT AN OUTPUT DEVICE:        (M)
         M - MONITOR ONLY
         P - PRINTER AND MONITOR
```

Bild 24: Ersatzwerte wieder herstellen

```
****************************************
*                                      *
*           RESTORE DEFAULTS           *
*                                      *
****************************************
--RESTORE DEFAULTS--
     FOR SOURCE SLOT:  6
                DRIVE:  1
   DESTINATION SLOT:  6
                DRIVE:  2
 SELECT AN OUTPUT DEVICE:        M
         M - MONITOR ONLY
         P - PRINTER AND MONITOR
```

---

Bild 25: FILER verlassen

```
****************************************
*                                      *
*                 QUIT                 *
*                                      *
**PREFIX: /PEEKER/AUFSAETZE/***********
--QUIT AND LOAD--
   PATHNAME: (/STIEHL/BASIC.SYSTEM
                                      )
```

Die Marke der Profis

IWT SOFTWARE

# jane mit der Maus...

alles in einem...

für Apple II, //e, 2c
Steuerung mit der Apple-Maus, Arctronics-Maus, Joystick oder Koala-Pad,
Fremddateien wie z. B. Apple-Writer können gelesen und geschrieben werden.

**Janelist** ist ein Dateiprogramm, mit dem alle Arten von Informationen gespeichert, sortiert und wieder aufgefunden werden können. Neben zwei festen Datenformaten (je ein Adressbuch für private und geschäftliche Adressen) können weitere Formate durch den Anwender definiert werden. Natürlich ist auch der Ausdruck von Listen und Adressaufklebern möglich.

**Janecalc** ist ein Tabellenkalkulationsprogramm, vorwiegend für kaufmännische und finanztechnische Berechnungen. Neben den Grundrechenarten stehen Prozentrechnung, Summen-, Mittelwert-, Minimum-, Maximum- und Zählfunktionen zur Verfügung. Die einzelnen Felder des Arbeitsblattes können beliebig miteinander verknüpft werden. Bei Änderung eines Eintrages werden alle Ergebnisfelder automatisch neu berechnet.

**Janewrite** ist ein Textverarbeitungsprogramm mit allen wichtigen Eigenschaften und Funktionen, das so bedient wird wie eine normale Schreibmaschine. Verschiedene Schriftarten, hoch- und tiefgestellte Texte und Unterstreichungen können eingegeben werden und sind direkt auf dem Bildschirm sichtbar.

Jane ohne Maus, Best.-Nr. 702 01101, DM 595,– incl. MwSt.*
Jane mit Maus, Best.-Nr. 702 01102, DM 1250,– incl. MwSt.*
* unverbindliche Preisempfehlung

## ... und natürlich in deutsch

IWT Software Service – für Information, Wissenschaft, Technologie
Höhestraße 66, Postfach 1325, 5093 Burscheid 1, Tel. (0 21 74) 6 28 15, Tx 5213989 iwt
Vertrieb: Dahlienstr. 4, 8011 Vaterstetten/Baldham, Tel. (0 81 06) 3 10 17, Tx 5213989 iwt

IWT

1001 Microcomputer . . . . . 1599.00
NCS-1 Grundversion
Dual Prozessor 6502 + Z80
64 KB RAM, 16 KB ROM
freie Tastatur im IBM-Look

2002 SAKATA Disk-Drive . . . 499.00
35/40 Track 160 KB
Direktantrieb, Newtech Logic
Shinon Mechanik

| Best.-Nr. | Bezeichnung | incl. MwSt. |
|---|---|---|
| 2009 | TOSHIBA DS/DD Doppeldrive für Apple, unformatiert 2 MB incl. Controler | 2399.00 |
| 4000 | PHILIPS Monitor TP200 18Mhz | 249.00 |
| 4004 | AGC Monitor bernstein 22Mhz, 1000 x 900 Lines Auflösung. | 349.00 |
| 5000 | SAKATA Matrixdrucker SP-1000, 100Z/sec 120 Z/z Centronics und Seriell 9 x 11 Matrix, Graphicfähig | 999.00 |
| 5001 | DWX 305 Typenraddrucker Parallelschnittstelle 22 Z/sec. Option Einzelblattzuführung. Qume Farbbänder und Typenräder. | 1299.00 |

| Best.-Nr. | Bezeichnung | incl. MwSt. |
|---|---|---|
| 3301 | Super Serial Card | 229.00 |
| 3302 | 7710 Asyn. Serial Interface | 279.00 |
| 3303 | Communication Card | 149.00 |
| 3304 | Rs-232 Card | 119.00 |
| 3305 | 16 K RAM Card ohne Kabel | 109.00 |
| 3306 | 16 K RAM Card mit Kabel | 119.00 |
| 3307 | 128 K RAM Card mit Software | 488.00 |
| 3308 | Z-80 Card Standard | 104.00 |
| 3309 | Z-80 Card Plus (NEW) | 119.00 |
| 3310 | Z-80 B mit 64 K RAM | 799.00 |
| 3312 | 80 Z-Card mit Softswitch | 169.00 |

| Best.-Nr. | Bezeichnung | incl. MwSt. |
|---|---|---|
| 3313 | 80 Z-Card mit Softswitch und Software | 299.00 |
| 3314 | Music Card | 179.00 |
| 3315 | Clock Card | 169.00 |
| 3316 | Wild Card (Copierkarte) | 119.00 |
| 3317 | Eprom Writer 2716-27128 | 199.00 |
| 3318 | Eprom Writer 2716-2764 | 149.00 |
| 3321 | Disk-Control-Card 13/16 S | 109.00 |
| 3322 | Epson Printer-Card m. Kabel | 129.00 |
| 3323 | Grappler Card (graphic) | 139.00 |
| 3324 | Grappler Card Plus | 179.00 |
| 3325 | Grappler Card m. 16 K RAM | 399.00 |
| 3326 | RGB Card (TTL out) | 199.00 |
| 3327 | RGB Card (linear out) | 199.00 |
| 3328 | 6522 Prozessor Card | 129.00 |
| 3329 | 6809 Prozessor Card | 449.00 |
| 3330 | IEEE-488 Card mit Kabel | 389.00 |
| 3331 | 8 bit 8 Ch AD-DA Wandler | 399.00 |
| 3332 | 8 bit 16 Ch AD-DA Wandler | 799.00 |
| 3333 | Parallel Card | 129.00 |
| 3334 | Super Parallel Card | 299.00 |
| 3337 | 6502 CPU 48 KB Motherboard | 699.00 |
| 3338 | 6502 CPU 64 KB Motherboard | 739.00 |
| 3339 | Dual CPU 6502 und Z-80 64 KB Motherboard | 749.00 |
| 3340 | Pal Card | 109.00 |
| 3342 | Joystick autocenter | 89.00 |

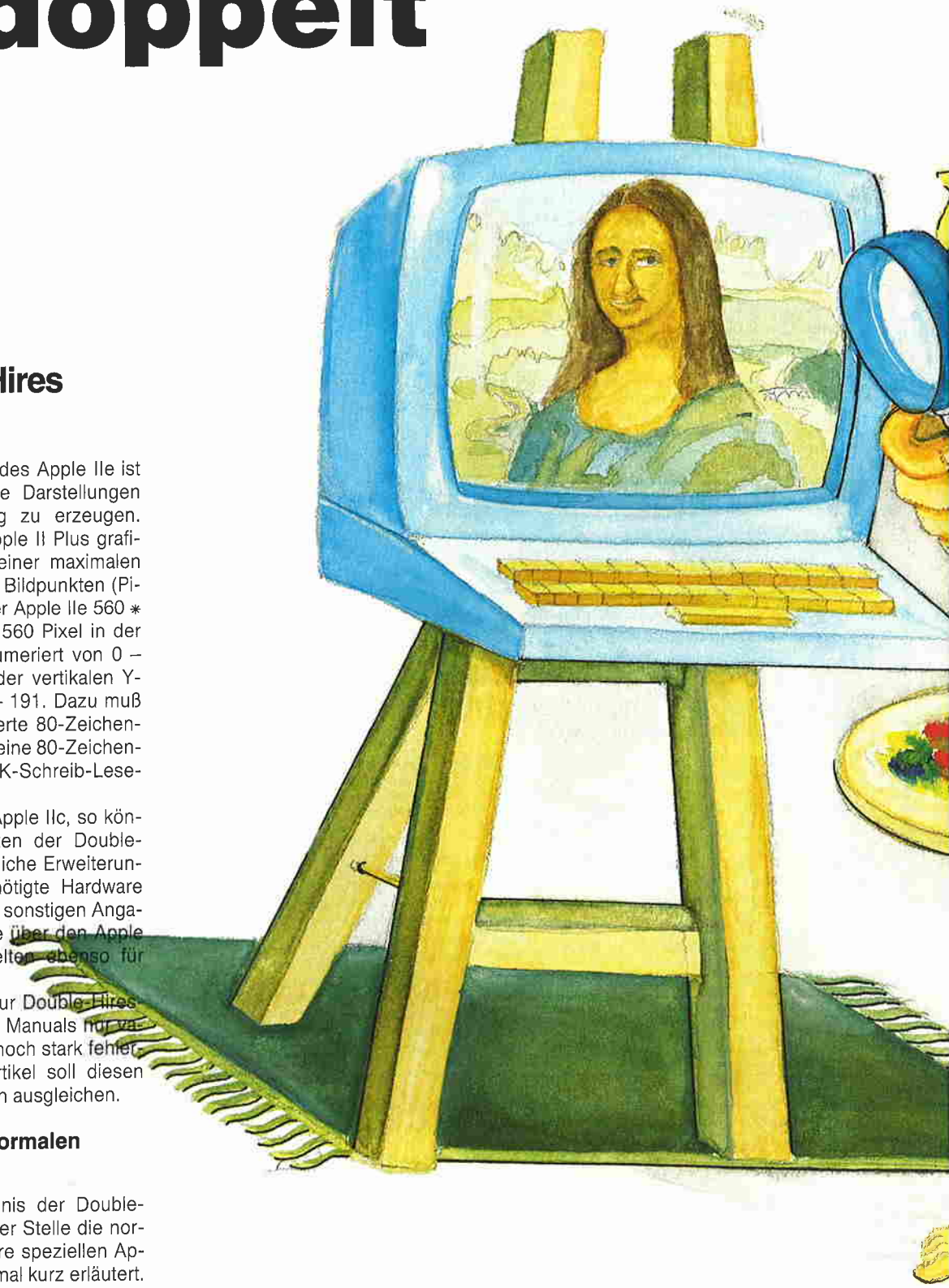
Preise verstehen sich incl. 14% Mehrwertsteuer. Weitere Informationen direkt von uns gegen DM 1.30 Rückporto.
**Händleranfragen auch aus dem Ausland erwünscht** sowie Handelsvertretung gesucht.

**Generalimporteur**
**Vertrieb**
**Service**

**Nettetaler Computersysteme GmbH**

Klemensstraße 7 · D 4054 Nettetal 2
Kaldenkirchen · Telefon 0 21 57 / 10 67
Telex 17 215 732 · Teletex 215 732 = NCS

# Wie man die Grafik verdoppelt



## Teil 2: Double Hires

**von Karl-Walter Bott**

Ein interessanter Aspekt des Apple IIe ist seine Fähigkeit, grafische Darstellungen hoher Bildpunktauflösung zu erzeugen. Während Apple II und Apple II Plus grafische Darstellungen mit einer maximalen Auflösung von 280 * 192 Bildpunkten (Pixel) unterstützten, kann der Apple IIe 560 * 192 Punkte ansprechen, 560 Pixel in der horizontalen X-Achse, numeriert von 0 – 559 sowie 192 Pixel in der vertikalen Y-Achse, numeriert von 0 – 191. Dazu muß im Apple IIe eine erweiterte 80-Zeichenkarte installiert sein, d. h. eine 80-Zeichenkarte mit zusätzlichem 64K-Schreib-Lesespeicher.
Sind Sie Besitzer eines Apple IIc, so können Sie die Möglichkeiten der Double-Hires-Grafik ohne zusätzliche Erweiterungen nutzen, da die benötigte Hardware bereits eingebaut ist. Alle sonstigen Angaben in diesem Artikel, die über den Apple IIe gemacht werden, gelten ebenso für den Apple IIc.
Die Firma Apple macht zur Double-Hires-Grafik in ihren Reference Manuals nur vage Angaben, die zudem noch stark fehlerbehaftet sind. Dieser Artikel soll diesen Mangel an Dokumentation ausgleichen.

### 1. Organisation der normalen Hires-Grafik

Zum besseren Verständnis der Double-Hires-Grafik wird an dieser Stelle die normale Hires-Grafik und ihre speziellen Applesoft-Befehle noch einmal kurz erläutert.

Der Apple IIe kann im normalen HGR-Modus Bilder mit einer maximalen Auflösung von 280 * 192 Bildpunkten erzeugen. Wenn wir uns auf eine schwarzweiße Bilddarstellung beschränken, kann jeder Bildpunkt auf dem Monitor genau zwei Zustände einnehmen, entweder sichtbar oder unsichtbar. Zu jedem Bildpunkt auf dem Monitor existiert ein entsprechendes Bit im Bildschirmspeicher, welches den Zustand des Bildpunktes definiert, wobei der binäre Wert 1 für sichtbar und der binäre Wert 0 für unsichtbar steht. Wir benötigen also bei einer Auflösung von 280 * 192 Bildpunkten insgesamt 53760 Bits, um jeden Punkt auf dem Bildschirm eindeutig zu definieren.
Der Apple IIe stellt hierfür wahlweise zwei zusammenhängende Speicherbereiche von 8K Größe zur Verfügung:

HGR-Seite 1: $2000-$3FFF (HGR)
HGR-Seite 2: $4000-$5FFF (HGR2)

Werfen wir einen kurzen Blick auf **Diagramm 1**. Wir erkennen folgendes: Jeweils 8 Bits sind zu einem Byte zusamengefasst, wobei nur 7 zur Darstellung von Bildpunkten verwendet werden. Das Bit 7 erfüllt eine besondere Funktion, die später erläutert wird. Zur Abbildung einer Grafikzeile mit 280 Bildpunkten sind demnach genau 40 Bytes notwendig. Kompliziert wird die Sache dadurch, daß die Grafikzeilen nicht in aufsteigender Reihenfolge dem Grafik-Bildschirmspeicher zugeordnet sind.

Im normalen Textmodus werden 40 (Text-) Spalten dargestellt, die ihrerseits wieder aus 7 Bildpunktspalten bestehen. Somit erhalten wir 40 * 7 = 280 Bildpunktspalten, da jeweils 1 Byte nur 7 von 8 Bits direkt zur Grafikanzeige verwendet.
Vertikal erhalten wir 24 Textzeilen, die 24 * 8 = 192 Bildpunktreihen entsprechen.

Die Anfangsadresse der 1. Grafikzeile beginnt bei der HGR-Seite 1 bei $2000 und endet bei $2027.
Die Adressenzuordnung bei HGR-Seite 2 erfolgt analog zur HGR-Seite 1, jedoch mit der Anfangsadresse $4000 statt $2000.

Es fehlt nun noch die Erklärung zum Bit 7, welches nicht direkt zur Anzeige eines Bildpunktes verwendet wird. Es steht in Verbindung mit der Fähigkeit des Apple IIe, hochauflösende Grafiken farbig darzustellen. Grundsätzlich wird jeder Punkt auf dem Bildschirm farbig dargestellt. Sind Sie glücklicher Besitzer eines Farbmonitors, dann probieren Sie folgendes aus:

```
HGR:HCOLOR=3:REM Weiß
```

Danach zeichnen wir mit

```
HPLOT 10,0 TO 10,100
HPLOT 33,0 TO 33,100
```

2 Linien, von denen die eine in violetter Farbe und die andere in grüner Farbe erscheint. Allgemein gilt, daß alle Punkte mit gerader X-Koordinate in violetter Farbe abgebildet werden, während alle mit ungerader X-Koordinate grün erscheinen.
Wenn zwei nebeneinander liegende Punkte mit unterschiedlicher Farbe gleichzeitig gesetzt sind, erhält man die Farbe Weiß. Es ergeben sich daraus die vier Farben Schwarz, Grün, Violett und Weiß.
Ist das Bit 7 gesetzt, so kommt der Farbensatz Schwarz, Orange, Blau und Weiß hinzu. Man hat also insgesamt sechs Farben zur Verfügung. Es kann aber gleichzeitig in einem Byte nur ein Farbensatz verwendet werden, entweder Schwarz, Grün, Violett, Weiß oder Schwarz, Orange, Blau und Weiß.

Die folgenden Versuche sollen Ihnen die Organisation der normalen schwarzweißen Hires-Grafik verdeutlichen.

Für unsere Versuche wählen wir die HGR-Seite 1, und zwar den Modus Grafik mit Text. Geben Sie nun folgende Befehle ein:

```
HGR: TEXT: HGR
CALL –151
```

Wir befinden uns im Monitor und geben nunmehr

```
2000:01
```

ein und sehen auf dem Bildschirm in der linken oberen Ecke einen Punkt. Angenommen wir möchten in der 0. Grafikzeile die Pixel 0 ... 13 einschalten, so geben wir folgendes ein:

```
2000:7F
2001:7F
```

und sehen einen waagrechten Strich am Anfang der 0. Grafikzeile. Geben wir nun statt $7F (%01111111) etwa $55 (%01010101) und $2A (%00101010) ein, und zwar so

```
2000:55
2001:2A
```

dann sehen wir eine unterbrochenen waagrechte Linie, da jetzt nur jedes zweite der 14 Pixel eingeschaltet ist.

## 2. Normale Hires-Befehle

Im folgenden Abschnitt sollen die Applesoft-Hires-Grafikbefehle noch einmal kurz zusammengefasst werden. Ausführliche Informationen sind im Applesoft Reference Manual enthalten. Zum Verständnis der

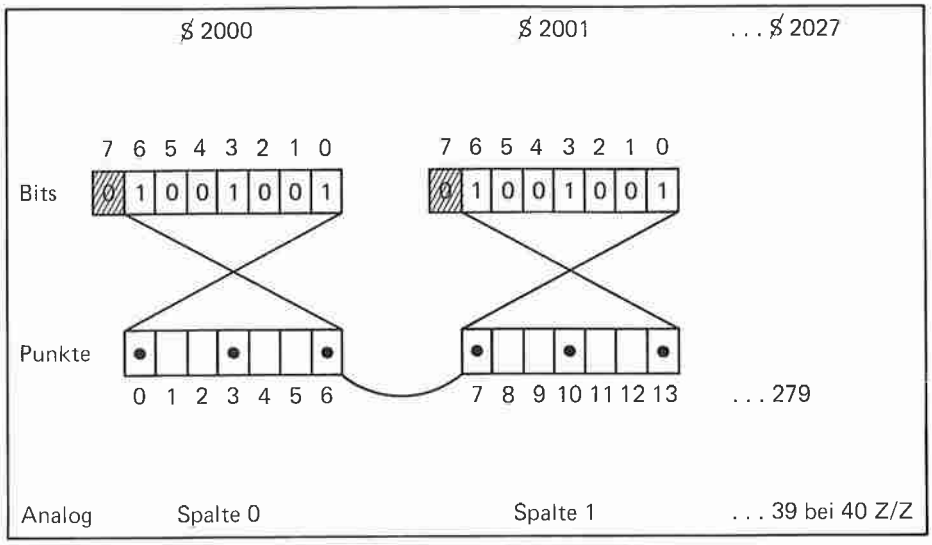



**Diagramm 1:** Zuordnung von internen Bitmuster und sichtbaren Bildpunkten bei einfacher Hires-Grafik (2000 : 49 49)

HPLOT-Befehle sei an dieser Stelle erwähnt, daß der Ursprung des Koordinatensystems, auf das sich die HPLOT-Befehle beziehen, der Position des oberen linken Bildschirmpunktes entspricht.

**HGR**

schaltet auf Hires-Grafik mit vier Textzeilen und schwarzer Farbe um.

**HGR2**

schaltet auf HGR-Seite 2 um. Hier ist nur volle Grafik ohne Text möglich.

**HCOLOR = F,** z. B. HCOLOR = 3

selektiert die gewünschte Farbe, wobei F zwischen 0 und 7 liegen muß.

**TEXT**

schaltet den Hires-Grafikmodus wieder ab.

**HPLOT X, Y,** z. B. HPLOT 100, 100

plottet einen Punkt mit der Koordinate X, Y.

**HPLOT X1, Y1 TO X2, Y2 ... TO Xn,Yn**

plottet eine Linie, die zwei Koordinaten verbindet.

**DRAW N AT X, Y,** z. B. DRAW 1 AT 10, 70 plottet eine Linienfolge (Shape) mit der Nummer N in der Shape-Tabelle an der Stelle X, Y

**XDRAW N AT X, Y**

entspricht DRAW, jedoch erfolgt die Darstellung nicht in der zuvor mit HCOLOR festgelegten Farbe, sondern mit der Farbe, die sich zu der auf dem Bildschirm sichtbaren komplementär verhält, wobei komplementär im Sinne der zwei möglichen Farbensätze gemeint ist (siehe Applesoft Referene Manual).

**SCALE = N**

bestimmt den Vergrößerungsfaktor, mit dem eine Linienfolge dargestellt wird, wobei N zwischen 0 und 255 liegen muß.

**ROT = N**

bestimmt den Drehwinkel bezüglich des Mittelpunktes der Linienfolge. Der Wert von N muß im Bereich zwischen 0 und 255 liegen.

## 3. Double-Hires-Bildschirm

Wir erinnern uns, daß wir zur Definition aller Bildschirmpunkte im Hires-Grafikmodus einen zusammenhängenden Speicherbereich von 8K benötigten. Die Double-Hires-Grafik ermöglicht eine Auflösung von 560 * 192 Bildpunkten, was 107520 Bits oder 16K Bildschirmspeicher erfordert, wenn wir die Systematik der Zuordnung Bit = Bildpunkt wie bei der normalen Hires-Grafik beibehalten. Woher kommen nun die fehlenden 8K Bildschirmspeicher? Die erweiterte 80-Zeichenkarte stellt weitere 64K Schreib-Lesespeicher zur Verfügung, von denen sich 8K ebenfalls zur Grafikdarstellung verwenden lassen. Der Double-Hires-Bildschirmspeicher besteht nunmehr aus zwei Teilen:

a) dem unteren Bereich von $2000 ... $3FFF auf dem 64K-Motherboard
b) dem oberen Bereich von $2000 ... $3FFF auf der 64K-Karte

Im Gegensatz zur normalen Hires-Grafik sind jetzt 80 Bytes zur Darstellung einer vollen Grafikzeile notwendig, wobei jeweils 40 Bytes aus dem „oberen" und 40 Bytes aus dem „unteren" Speicherbereich stammen. Welche Bildschirmpunkte aus welchem Speicherbereich kommen, illustriert **Diagramm 2.**

Aus dem „oberen" Speicherbereich stammen die geraden Spalten (0, 2, 4, 6 ... 78), aus dem „unteren" Bereich die ungeraden Spalten (1, 3, 5, 7 ... 79), wobei jede Spalte aus 8 Bits besteht, von denen 0 bis 6 ausgewertet werden, während das Bit 7 wieder die Farbinformation beinhaltet. Das bedeutet, daß die Bildschirmpunkte 0 ... 6 von der Karte, 7 ... 13 aus dem Hauptspeicher, und 14 ... 21 von der Karte stammen usw.
Da die beiden Speicherseiten den identischen Adreßraum belegen, muß ein Schalter existieren, der bestimmt, ob in die „untere" oder in die „obere" Page geschrieben werden soll, damit jeder beliebige Bildschirmpunkt des Double-Hires-Grafikspeichers unabhängig von anderen gesetzt oder gelöscht werden kann. Der Softswitch
$C055 (= Page 2 = „oben")
selektiert den „oberen" Teil des Grafikspeichers, während der Softswitch
$C054 (= Page 1 = „unten")
den „unteren" Teil aktiviert. Der nächste Versuch verdeutlicht die Wirkung der Softswitches.

Zunächst schalten wir mit
HGR: TEXT: HGR
auf Hires-Grafik mit Text. Anschließend aktivieren wir die erweiterte 80-Zeichenkarte mit
PR#3
und gehen erneut mit
CALL –151
in den Monitor und geben
C05E:0
ein.

Der Softswitch $C05E schaltet einen der vier Standard-TTL-Ausgänge, Annunciator 3 am Spieleausgang (Game-I/O), „aus".

# MATTES® COMPUTERSYSTEME

Alle Preise verstehen sich in DM incl. gesetzl. Mwst. *) Apple ist ein Warenzeichen der Apple Comp. Inc.

7470 ALBSTADT 2 · Theophil-Wurm-Straße 7 Postfach 1549

**BROTHER**
**Typenradschreibmaschine/**
**-drucker CE50**
**mit Centronicsinterface**
**Kabel mitbestellen!**

**1148.–**

**vom Computer aus ansteuerbar sind z.B. folgende Funktionen:**

**automatisches Unterstreichen**
**variabler Zeichen-/Zeilenabstand**
**Halbzeilenvorschub-/rückzug**
**zentrieren**
**Sperrschrift**
**setzen von Rändern und Tabulatoren u.v.m.**

| | |
|---|---|
| **Brother CE50 ohne Interface** | **925.—** |
| **Umbau einer "alten" CE50 mit MAT-Interface** | **248.—** |

**Apple* kompatible Rechner und Peripherie**

| | |
|---|---|
| HM Duo 64kB, Z-80/6502 | 1198.— |
| Sherry 64kB, 6502/Z-80 mit abgesetzter Tastatur, 10er Block... | 1499.— |
| Disk-Laufwerk 143 kB -Apple* kompatible- aus Deutscher Produktion | 599.— |
| dito mit Controller | 699.— |
| **320kB** Laufwerk -voll Apple* kompatible- | 840.— |
| **640kB** Laufwerk | 950.— |
| Z-80 Karte | 149.— |
| CP/M 3.0 Original Handbücher | 69.— |
| EPROM Brenner 2708 - 27256 | 299.— |
| 80 Zeichenkarte mit Softswitch | 199.— |

**Drucker**

| | |
|---|---|
| Epson FX 80 | 1598.— |
| STAR gemini 10x | 999.— |
| STAR delta 10 | 1449.— |
| STAR radix 10 | 1999.— |
| STAR radix 15 | 2399.— |
| Grafikdruckerinterface | 199.— |
| Abgesetzte Funktionstastatur, frei programmierbar, vom Computer aus ladbar mit 10er Block IBM Design | 399.— |

**Software**

| | |
|---|---|
| Copy Killer (siehe Peeker Nr. 1) | 228.— |
| Wordstar 3.3 deutsch | 1250.— |
| DBase II | 1299.— |
| Multiplan | 770.— |
| DBase Compiler | 3216.— |
| Datgen Datenbank Generator für Wordstar | 228.— |
| Compas 80 Pascal Entwicklungssystem | 995.— |
| SuperCalc | 599.— |
| SuperCalc 2 | 799.— |
| Update Supercalc nach Supercalc 2 | 300.— |

**Tel. 0 74 32/1 33 16 · Telex 763 317 mat d**

---

# weidemann electronic

# Apple Service Händler Level I
# 24 Std. Reparaturservice

**Achtung:** alle Geräte mit Seriennummer vom Hersteller

## Apple II e

| | |
|---|---|
| Apple IIe mit 64 kb | 2418,– DM |
| **Einstiegspaket** (IIe, Monitor, Disk mit Contr.) | 3368,– DM |
| Disk m. C. (Prodos, DOS) | 1177,– DM |
| Disk o. Contr. | 866,– DM |
| DuoDisk incl. Contr. | 2016,– DM |
| Profile 5 MB incl. Interf. | 4299,– DM |
| II MOuse incl. Interface | 376,– DM |

## Apple IIc

| | |
|---|---|
| Apple IIc | 2655,– DM |
| **Komplettpaket** (IIc incl. Monitor und Ständer) | 3249,– DM |
| IIc 2. laufwerk | 855,– DM |
| IIc Mouse incl. Mousepaint | 262,– DM |

## Drucker + Plotter

| | |
|---|---|
| Apple Scribe incl. Kabel | 849,– DM |
| Apple Imagewriter 12" incl. Kabel | 1484,– DM |
| Apple Imagewriter 15" incl. Kabel | 2176,– DM |
| Apple A3 Plotter 4 Farben | 2252,– DM |
| Epson RX-80 | 957,– DM |
| Epson FX-80 | 1479,– DM |
| Epson FX-100 | 1917,– DM |
| Parallel Interface incl. Kabel | 211,– DM |
| Seriell Interf. für Apple Drucker | 211,– DM |
| parallel/seriell Interface | 229,– DM |

## Macintosh

| | |
|---|---|
| Macintosh 128 kb | 6299,– DM |
| Macintosh 128 kb (110 V. Vers.) | 5999,– DM |
| Macintosh 512 kb | 9188,– DM |
| Aufrüstung auf 512 kb | 2948,– DM |
| Externes Laufwerk | 1239,– DM |

Fordern Sie unsere Softwareliste an!

## Monitore + Karten

| | |
|---|---|
| Apple IIe Monitor | 562,– DM |
| Apple IIc Monitor | 512,– DM |
| **Zenith ZVM-122-EA bernstein** | **299,– DM** |
| Zenith ZVM-123-EA rün | 307,– DM |
| Taxan RGB Vision I, 15 MHz. | 755,– DM |
| Taxan RGB Vision II, 18 MHz. | 1073,– DM |
| Taxan RGB Vision III, 20 MHz. | 1368,– DM |
| 80 Zeichenkarte IIe | 171,– DM |
| 80 Zeichenkarte + 64 kb | 376,– DM |
| RGB + 80 Zeichenkarte | 399,– DM |
| RGB + 80 Zeichen + 64 kb | 707,– DM |

## Software

**Wir liefern jede Software für Apple**
z.B. Aztek C-Compiler für Macintosh
**1699,– DM**
**Fordern Sie unsere Listen an!**

**Auf alle Apple Produkte 1 Jahr Garantie**
Sonstige 6 Monate (Epson ohne Druckkopf)
**Alle Preise incl. Mehrwertsteuer**

**Anfragen + Bestellungen**

Tel. 0 26 34 / 36 36 / 36 37 / 36 38 / 36 39
Telex: 86 85 13
IMCA/Auge Mailbox an U. Weidemann
5455 Rengsdorf Wohnplatz Weidemann

Dies hat zur Folge, daß jetzt sowohl die „untere" als auch die „obere" Hälfte des Double-Hires-Grafikspeichers gleichzeitig zur Anzeige kommen.
Auf dem Bildschirm werden wir eventuell ein schachbrett-ähnliches Muster sehen, weil wir durch den Befehl HGR nur den Teil des Double-Hires-Grafkspeichers löschen, der sich auf dem 64K-Motherboard befindet.
Um auch den zweiten Teil des Grafikspeichers zu löschen, der sich auf der 64K-Karte befindet, müssen wir mit dem Softswitch für Page 1 den oberen Speicherbereich auswählen und den HGR-Befehl noch einmal ausführen. Dazu geben wir noch folgendes ein:
C055:0 N F3F2G
Der Double-Hires-Bildschirm ist durch den direkten HGR-Aufruf „F3F2G" jetzt vollständig gelöscht.
Zur Vertiefung der erworbenen Kenntnisse über das Zusammenwirken von Haupt- und Zusatzspeicher machen wir noch folgenden Versuch. Wir geben ein:
C055:0 N 2000:7F 7F

Wir sehen auf dem Bildschirm eine horizontale Linie in den Spalten 0 und 2. Durch Ansprechen des Softswitch $C055 haben wir den Speicherbereich auf der 64K-Karte ausgewählt und durch die Bitmaske %01111111 ($7F) die Bildschirmpunkte 0 ... 6 und 14 ... 21 eingeschaltet. Gibt mann jetzt
C054:0 N 2000:7F 7F
ein, so sieht man ebenfalls eine horizontale Linie, und zwar in den Spalten 1 und 3. Wir haben durch $C054 den Teil des Double-Hires-Grafikspeichers angesprochen, der auf dem Motherboard liegt. Der Befehl aktiviert nunmehr die Bits 7 ... 13 und 21 ... 27.

Wie schon erwähnt, beinhaltet eine Grafikzeile im Double-Hires-Modus insgesammt 560 Punkte gegenüber dem normalen Hires-Modus mit 280 Bildpunkten. Die Firmware des Apple IIe (und auch des IIc!, Anm. der Red.) stellt jedoch keine speziellen Double-Hires-Grafikbefehle oder Assembler-Routinen zur Verfügung, die das Softswitch-Managment übernehmen. Um das Betätigen der Softswitches muß sich der Programmierer selbst kümmern. Die Assembler-Befehle

LDA $C050 (Grafik)
LDA $C057 (Hires)
STA $C001 (80-Store)
STA $C00D (80-Column)
LDA $C05E (AN3 aus = Double Hires an)

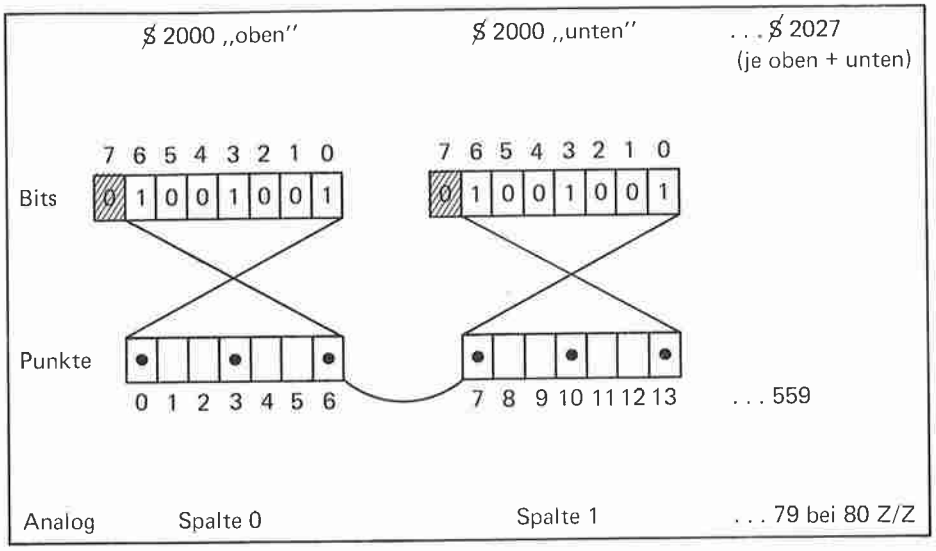



**Diagramm 2:** Zuordnung von internen Bitmustern und sichtbaren Bildpunkten bei doppelter Hires-Grafik (erst oben CO55 : O N 2000 : 49, dann unten CO54 : O N 2000 : 49)

schalten den Double-Hires-Modus ein. Danach kann mit

LDA $C055 (Page 2 = oben)
LDA $C054 (Page 1 = unten)

jeweils zwischen den zwei Bildschirmhälften mit dem identischen Adreßraum $2000-$3FFF umgeschaltet werden.

Jetzt können wir noch wahlweise Grafik mit vier Textzeilen oder volle Grafik einschalten:

LDA $C053 (Mixed Modus ein)
LDA $C052 (Mixed Modus aus = volle Grafik ein)

An dieser Stelle soll noch einmal auf die besondere Wirkungsweise des Page-Schalters hingewiesen werden. Ist der Double-Hires-Modus eingeschaltet, so bestimmt der Page-Schalter, ob in den Hauptspeicher oder in den Zusatzspeicher geschrieben werden kann, während im normalen Hires-Modus der Page-Schalter die beiden Grafikseiten $2000-$3FFF und $4000-$5FFF selektiert. Deshalb ist die Double-Hires-Grafik auf der HGR-Seite 2 $4000-$5FFF technisch nicht möglich.

Das Betätigen der Softswitches Hires und Mixed ist nur wirksam, wenn der Grafikmodus eingeschaltet ist (Text aus). Zur Aktivierung der Double-Hires-Grafik muß der Annunicator 3 *aus*geschaltet werden, *nicht* ein, wie im Apple-IIe-Reference-Manual fälschlicherweise angegeben.

Die LDA-STA-Konvention muß beim Ansprechen der Softswitches eingehalten werden, wobei die Assembler-Befehle LDA dem PEEK-Befehl und STA dem POKE-Befehl im Applesoft-Basic entsprechen.

Um die Double-Hires-Grafik wieder auszuschalten, müssen folgende Softswitches aktiviert werden:

LDA $C054 (Page 1)
LDA $C051 (Text)
LDA $C05F (AN3 an = Double Hires aus)

STA $C000 (40-Store)
STA $C00C (40-Column)

Die beiden letzten Softswitches dürfen nicht betätigt werden, wenn vor dem Einschalten des Double-Hires-Modus die 80-Zeichenkarte bereits eingeschaltet war, da sonst die ROM-Routinen der 80-Zeichenkarte nicht mehr korrekt arbeiten.
Sollten Ihre Experimente mit der Double-Hires-Grafik auf Ihrem Apple IIe nicht erfolgreich verlaufen sein, kann dies folgende hardware-technische Gründe haben:

– Sie besitzen keinen Apple IIe mit B-Motherboard (d. h. bei der Seriennummer auf der Hauptplatine steht am Ende kein B).
– Sie haben die Drahtbrücke, die Sie mit der erweiterten 80-Zeichenkarte erhalten haben, nicht installiert (siehe Ergänzungsblatt zur 80-Zeichenkarte). Bei neueren

Ausführungen und manchen Nachbauten der erweiterten 80-Zeichenkarte muß keine Drahtbrücke installiert werden.
– Sie haben eine Nachbau-80-Zeichenkarte, die nicht über die technischen Möglichkeiten zur Darstellung von Double-Hires-Grafik verfügt.
(Wenn Sie im Monitor mit Double Hires experimentieren, so beachten Sie, daß die 80-Zeichen-Firmware die Softwitches oft aufhebt. Um bei Experimenten ganz sicher zu gehen, schreibe man sich am besten ein kleines Programm oder lade INTEGER-Basic einschließlich des alten Monitors in die Language Card und bleibe bei 40 Z/Z. Anm. der Red.)

## 4. Hinweise zu den Programmen

### 4.1. DHGR.APSOFT.DEMO

Das Demonstrationsprogramm zeigt, wie man die Double-Hires-Softswitches aus einem Applesoft-Programm richtig schaltet. Das Programm verdeutlicht die Problematik, die entsteht, wenn man im Double-Hires-Modus die normalen Hires-Routinen des Applesoft-Interpreters verwenden will. Man erkennt, daß jeweils zwischen Page 1 und Page 2 umgeschaltet werden muß, da die ROM-Grafik-Routinen bekanntlich nicht die obere Bildschirmhälfte anwählen können.
Die Applesoft-Grafikbefehle können im allgemeinen nicht mehr direkt verwendet werden, so daß zunächst eine Routine in Applesoft oder Assembler geschrieben werden muß, um aus den Koordinaten des Punktes jeweils die richtige Page auszuwählen. Diese Routine ist in den Zeilen 900–940 realisiert. Auf dem Bildschirm erscheint eine Sinuskurve, die mit einer Auflösung von 560 * 192 geplottet wird.

### 4.2. AMPER.DOUBLE.HIRES (Applesoft und Assembler)

Dieses Programm zeigt eine weitere Möglichkeit, Double-Hires-Grafik zu realisieren. Wir erinnern uns, daß eine Double-Hires-Zeile 560 Pixel umfaßt, der HPLOT-Befehl aber nicht zwischen der oberen und unteren Seite des Double-Hires-Bildschirms auswählen kann. Das Betätigen der Softswitches übernimmt jetzt ein Assembler-Programm. Der Applesoft-Befehl „HPLOT" wird hier nicht mehr verwendet, was eine spürbare Steigerung der Geschwindigkeit der Plotfunktion im Double-Hires-Modus zur Folge hat. Außerdem demonstriert das Programm, wie man eine Assembler-Routine mit Hilfe des Ampersand-Befehls aus einem Applesoft-Programm aufruft.
Mit dem Befehl „& X, Y" kann jeder Punkt aus einem Applesoft-Programm heraus auf dem Double-Hires-Bildschirm geplottet werden, ohne daß man sich um die Softswitches kümmern muß.

An dieser Stelle soll die Struktur des Assembler-Programms kurz erläutert werden:
In den Zeilen 23–28 wird der Ampersand-Vektor „gelinkt", d. h. er zeigt jetzt nicht mehr auf einen „RTS"-Befehl im ROM, sondern direkt auf die Startadresse des Plot-Programmes.
Die Zeilen 30–60 bewirken das Umschalten auf volle Double-Hires-Grafik ohne Text und die Rückkehr in das Applesoft-Programm.

Das korrekte Abschalten des Double-Hires-Modus wird in dem Unterprogramm in den Zeilen 38–43 erreicht.
Danach werden in den Zeilen 48–60 sowohl die untere als auch die obere Grafikseite gelöscht, was dem HGR-Befehl im normalen Hires-Modus entspricht.
In der Zeile 72 beginnt der eigentliche Teil des Programms. Es werden die Plotparameter aus dem aufrufenden Basic-Programm geholt.
Aus der Tabelle, die in den Zeilen 111–164 definiert ist, wird die der Bilschirmkoordinate zugeordnete Adresse im Bildschirmspeicher ermittelt. Nach Aufruf des Divisons-Unterprogramms in Zeile 89 wird in den Zeilen 91–93 die Bitmaske zum Setzen des entsprechenden Bits aus der Tabelle in den Zeilen 168–182 ermittelt.
Ab Zeile 95 wird die obere oder die untere Grafikseite ausgewählt, je nachdem, ob sich bei einer Division der X-Koordinate durch 7 ein ungerader oder ein gerader Divisionsrest ergeben hat.
In den Zeilen 102–106 wird dann das Bit im Bildschirm-RAM gesetzt, das den entsprechenden Bildschirmkoordinaten zugeordnet ist, und die Rückkehr zum Applesoft-Programm veranlaßt. Dieses Programm namens **AMPER.DOUBLE.HIRES.BAS** erklärt sich im großen und ganzen selbst:

In Zeile 100 wird die 80-Zeichenkarte mit „PRINT CHR$ (21)" korrekt abgestellt und der Assembler-Teil des Programms geladen.
Die Zeilen 120–130 schalten auf volle Double-Hires-Grafik um und löschen den gesamten Bildschirm.
In den Zeilen 180 und 190 werden dann die einzelnen Punkte der 2 Parabeln geplottet.
Zum Schluß schaltet „CALL 24579" in der Zeile 200 den Double-Hires-Modus wieder ab.

### 4.3. DHGR.LINEPLOTTER

Dieses Programm veranschaulicht, wie man auf dem Double-Hires-Grafikschirm Linien aus einem Applesoft-Programm heraus zeichnen kann. Es ist ähnlich wie das Programm DHGR.APSOFT.DEMO aufgebaut. Interessant sind die Zeilen 400–498, die es ermöglichen, zwei Punkte auf dem Double-Hires-Grafikschirm durch eine Linie zu verbinden.
Das Programm verdeutlicht die Probleme, die entstehen, wenn man gänzlich auf Applesoft-Routinen verzichten muß, da der Basic-Befehl „HPLOT X1, Y1 TO X2, Y2" ausscheidet, weil er nicht zwischen den beiden Grafikseiten des Double-Hires-Bildschirms alternieren kann, was beim Plotten von schrägen Linien notwendig ist.
Das Programm zeichnet ein Quadrat mit seinen zwei Diagonalen im Double-Hires-Modus.
Es sei jedoch nicht verschwiegen, daß der in Applesoft formulierte Algorithmus relativ langsam ist, so daß es sich empfiehlt, ein geeignetes Programm in Assembler zu entwickeln.

## 5. Schlußbemerkung

Mit der Double-Hires-Grafik werden die Grafikfähigkeiten des Apple IIe und IIc erheblich gesteigert, was diesen Computern mit Sicherheit neue Anwendungsgebiete erschließen wird, auch wenn z. B. für CAD-Anwendungen die vertikale Auflösung von 192 Bildpunkten doch nicht ganz ausreichend erscheint.
Hinderlich ist jedoch, daß die normalen Applesoft-Grafikbefehle nur noch teilweise verwendet werden können. Auch andere Programmiersprachen wie z. B. PASCAL unterstützen die Double-Hires-Grafik nicht.
Der Artikel konnte auch nicht alle Aspekte der Double-Hires-Grafik behandeln, so z. B. Farbe oder Shapes bei Double Hires.
Die vorgestellten Programme und Experimente sollen dem Leser einen Einblick in die Double-Hires-Grafik verschaffen und vielleicht zur Entwicklung eigener Programme anregen.

DHGR.APSOFT.DEMO

```
100 REM Double-Hires-Applesoft-Demo
110 PRINT  CHR$ (21): REM 80-Zeichenkarte abstellen
120 CE = 49165: REM 80COL ein
130 CA = 49164: REM 80COL aus
140 HE = 49239: REM HIRES ein
150 HA = 49238: REM HIRES aus
160 TE = 49233: REM TEXT ein
170 TA = 49232: REM TEXT aus
180 ME = 49235: REM MIXED ein
190 MA = 49234: REM MIXED aus
200 SE = 49153: REM 80STORE ein
210 SA = 49152: REM 80STORE aus
220 PE = 49237: REM PAGE2 ein
230 PA = 49236: REM PAGE2 aus
240 AE = 49247: REM AN3 ein
250 AA = 49246: REM AN3 aus
300 REM Double Hires initialisieren
305 HGR : REM untere Seite löschen
310 P =  PEEK (MA): REM Mixed aus
312 POKE SE,0: REM 80-Store ein
314 POKE CE,0: REM 80-Column ein
316 P =  PEEK (PE): REM Page 2 (obere  Seite)
350 CALL 62450: REM untere Seite löschen
360 P =  PEEK (AA): REM AN 3 aus (Double Hires)
400 REM Koordinatensystem zeichnen
410 HCOLOR= 3
420 HPLOT 0,95 TO 279,95: REM X-Achse untere Seite
430 P =  PEEK (PA): REM Page 1 (obere Seite)
440 HPLOT 0,95 TO 279,95: REM X-Achse obere Seite
500 REM Sinus zeichnen
520 PI = 4 *  ATN (1)
530 S = 2 * PI / 559
600 FOR X = 0 TO 2 * PI STEP S
610 I =  INT (X * 559 / (2 * PI))
615 GOSUB 900
620 Y =  INT (96 -  SIN (X) * 95)
630 HPLOT XX,Y
640 NEXT X
700 GET A$
800 REM Double Hires abschalten
810 P =  PEEK (PA): REM untere Seite einschalten
820 P =  PEEK (TE): REM Text einschalten
830 P =  PEEK (AE): REM Double Hires ausschalten
840 POKE CA,0: REM 40-Column
850 POKE SA,0: REM 40-Store
860 TEXT
870 END
890 REM X-Koordinate berechnen und auf richtige Seite
    schalten
900 U = I / 7:BYTE =  INT (U):Z = BYTE / 2:PG = Z -  INT
    (Z):XX =  INT (( INT (Z) + U - BYTE) * 7 + .5)
910 REM Seite wählen
920 P =  PEEK (PE): REM obere Seite einschalten
930 IF PG THEN P =  PEEK (PA): REM untere Seite
    einschalten
940 RETURN
```

AMPER.DOUBLE.HIRES.BAS

```
100 PRINT : PRINT  CHR$ (21): REM 80-Z-OFF
110 PRINT  CHR$ (4)"BLOAD AMPER.DOUBLE.HIRES"
120 CALL 24576: REM HOOK
130 CALL 24582: REM CLEAR
135 HCOLOR= 3
140 XE = 559:YE = 191:A = XE / 2:B = YE / 2:R = 95:F1 =
    45:F2 = 45: REM 2 Parabeln
150 FOR X = A - R TO A + R: IF X < 0 OR X > XE GOTO 200
160 H = R * R - (X - A) * (X - A): IF H <  = 0 GOTO 200
170 Y1 = B +  SQR (H) - F1:Y2 = B -  SQR (H) + F1
180 IF Y1 > 0 AND Y1 <  = YE THEN  & X,Y1
190 IF Y2 > 0 AND Y2 <  = YE THEN  & X,Y2
200 NEXT X: GET X$: CALL 24579: REM UNHOOK
```

```
                        ORG  $6000
                *
                * AMPER.DOUBLE.HIRES
                * ==================
                *
                * 192 * 560 Bit-Punkte
                *
                * (aus U. Stiehl, Apple
                * Assembler, 1984)
                *
                IND1     EQU  $CE
                IND2     EQU  $50
                GETNUM   EQU  $E746
                *
6000: 4C 0C 60  H6000    JMP  HOOK       ;24576
6003: 4C 2E 60  H6003    JMP  UNHOOK     ;24579
6006: 4C 3E 60  H6006    JMP  CLEAR      ;24582
                *
6009: 00        YACHSE   HEX  00
600A: 00        XACHSEL  HEX  00
600B: 00        XACHSEH  HEX  00
                *
600C: A9 4C     HOOK     LDA  #$4C
600E: 8D F5 03          STA  $3F5
6011: A9 62             LDA  #<AMPER
6013: 8D F6 03          STA  $3F6
6016: A9 60             LDA  #>AMPER
6018: 8D F7 03          STA  $3F7
                *
601B: AD 52 C0          LDA  $C052      ;NO-MIX
601E: AD 57 C0          LDA  $C057      ;HI-RES
6021: AD 50 C0          LDA  $C050      ;GRAPHIC
6024: 8D 01 C0          STA  $C001      ;80STORE
6027: 8D 0D C0          STA  $C00D      ;80COL
602A: AD 5E C0          LDA  $C05E      ;AN3
602D: 60                RTS
                *
602E: AD 56 C0  UNHOOK   LDA  $C056      ;LO-RES
6031: AD 51 C0          LDA  $C051      ;TEXT
6034: AD 54 C0          LDA  $C054      ;PAGE1
6037: 8D 00 C0          STA  $C000      ;800FF
603A: 8D 0C C0          STA  $C00C      ;40COL
603D: 60                RTS
                *
                * CLEAR $2000-$3FFF
                *
                *
603E: A9 00     CLEAR    LDA  #$00
6040: 85 CE             STA  IND1
6042: A9 20             LDA  #$20
6044: 85 CF             STA  IND1+1
6046: A0 00             LDY  #$00
6048: AD 55 C0  CLEAR1   LDA  $C055
604B: A9 00             LDA  #0
604D: 91 CE             STA  (IND1),Y
604F: AD 54 C0          LDA  $C054
6052: A9 00             LDA  #0
6054: 91 CE             STA  (IND1),Y
6056: C8                INY
6057: D0 EF             BNE  CLEAR1
6059: E6 CF             INC  IND1+1
605B: A5 CF             LDA  IND1+1
605D: C9 40             CMP  #$40
605F: 90 E7             BCC  CLEAR1
6061: 60                RTS
                *
                * GETNUM: 16-BIT IN $50-$51
                *          8-BIT IN X-REG
                *
                * & X-ACHSE, Y-ACHSE
                *
6062: 20 46 E7  AMPER    JSR  GETNUM
6065: 8E 09 60          STX  YACHSE
6068: A5 50             LDA  IND2
606A: 8D 0A 60          STA  XACHSEL
606D: A5 51             LDA  IND2+1
606F: 8D 0B 60          STA  XACHSEH
                *
6072: AC 09 60          LDY  YACHSE
6075: B9 AE 60          LDA  HGRADRL,Y
6078: 85 CE             STA  IND1
607A: B9 6E 61          LDA  HGRADRH,Y
607D: 85 CF             STA  IND1+1
                *
607F: AD 0A 60          LDA  XACHSEL
6082: 8D 40 62          STA  DIVDENDL
```

# WS 2000 WORLD STANDARD MODEM

## Die neue Version dieses weltweit benutzten professionellen Modems – immer noch zum unschlagbaren Preis von DM 798,–!

- ☆ Datenaustausch und Kommunikation mit praktisch jedem Computer weltweit möglich
- ☆ Zugriff zu Datenbanken, Mailboxen, Btx, Btx rückwärts, usw.
- ☆ Telex für alle durch einen neuen Dienst – mit Ihrem Computer und dem WS 2000 (fragen Sie uns)
- ☆ Alle gängigen Baudraten (75, 300/300, 600, 1200, 1200/75, 75/1200) und international üblichen Übertragung-Standards (CCITT, BELL) – umschaltbar per Hand oder per Computer (IC-Satz SK1 hierfür DM 96,90; Anschlußkabel UPL DM 48,–)
- ☆ Automatisches Wählen mit Zusatzplatine AD2 (DM 199,50) und Kabel UPL
- ☆ Automatisches Annehmen von Anrufen mit Zusatzplatine AA2 (DM 199,50)
- ☆ Einfacher Anschluß (parallel zur Telefonleitung) ; eingebautes Netzteil; deutsche Anleitung; 1 Jahr Garantie
- ☆ Viele Interfaces lieferbar; z.B.
  - CBM I für C64/VIC20 einschließl. Listing DM 136,80
  - AC Kommunikations-Karte für APPLE DM 330,60
  - SPEC für SPECTRUM einschließl. Software (auf ROM) DM 256,60
- ☆ Anschlußkabel zwischen Computer und Modem (bitte benötigten Steckertyp angeben) DM 57,–
- ☆ Liefermöglichkeit: sofort ab Lager Hamburg
- ☆ Alle Preise einschließlich MWSt. zuzüglich Verpackung, Porto und Nachnahme (Bei Vorauszahlung durch V-Scheck/Überweisung, Porto und Verpackung frei)

**Claus F. Erbrecht**
Computer Related Products
Lappenbergsallee 37 · 2000 Hamburg 20
Telefon 040/850 52 55
Bankverbindung: Bank für Gemeinwirtschaft (BLZ 200 101 11) Konto-Nr. 1241223700

**Achtung:** Nur für hausinterne Telefon-Anlagen und nicht amtsberechtigte Nebenstellen – in der BRD ist der Anschluß an das öffentliche Telefonnetz nicht gestattet!

# TERMINAL-KEYBOARD'S

Mod. 8

**NEU — CHERRY-KEYBOARD**
Professionelle neue, mod. Tastatur, 77 Tasten einschl. sep. 10er-Tastenfeld sowie vier Curser-Tasten und Home-Taste sowie Autorepeat. Tastenfarbe je nach Feld Weiß od. Grau. Tastenkopf mit aufgerauhter "Touch-Oberfläche". Kapazitive Tastenkontakte (leichtgängig mit Druckpunkt). Betr.-Spg. +5 V, ASCII-Parallel-Norm, betriebsbereite Ausführung: mit Staubschutzplatine, Basis-Platine Epoxy. Maße: 15,5 × 42,5 cm. Anschlußbelegung für Connector liegt bei, die beigefügte Interface-Karte ermöglicht auf APPLE-kompatible Computer direkte Cursersteuerung sowie alle gebräuchlichen Sondercodes.
**DM 149,–**

Mod. 9

**KEY-Tronic Keyboard Mod. 3-REV - F**
Professionelle neue moderne Tastatur, 108 Tasten, Tastenfarbe Schwarz/Grau mit weißer Beschriftung, normales **Tastenfeld, vier Cursertasten + HOME-Taste, sep. 10er Block** sowie weitere **29 Befehltasten**, betriebsbereites Keyboard mit SERIELLEM Ausgang ..... **DM 228,–**

Mod. 10 (Bild siehe Mod. 9)

## KEYTRONIC-Keyboard Mod. K-T-C

Neue, moderne, professionelle Tastatur mit **108 Tasten**, wovon **alle mit Funktionen** belegt sind, **voll textverarbeitungsfähig**, Groß + Kleinschreibung, moderne schwarze Tasten mit weißer Beschriftung, normales **Tastenfeld, sep. Cursertasten mit HOME-Taste, sep. 10er-Tastenfeld** sowie **29 Befehltasten, sämtliche Control-Zeichen, dir, Cursersteuerung, mit Autorepeat.** Betriebsbereites Keyboard **in ASCII-Parallel-Norm** einschl. Encoder-Interface, **Maße über alles:** 420×200 mm. Anschlußplan für Connector liegt bei. Durch Programmwahl über einen DIL-Schalter besonders individuelle Anpassung für folgende Computer-Typen: **BASIS-108, APPLE-II und APPLE-Compatible Computer, VT-100, TRS-80, ITT 2020 u.a.** sowie **Anwender mit ASCII-Parallel-Norm** per Stück **DM 278,–**

Mod. 11

## RAFI-KEYBOARD

mit **intelligentem INTERFACE in ASCII-Parallel-Norm** und seriell möglich, 57 Tasten frei programmierbar in 4 Ebenen, Maße 95×480 mm.

Das KEYBOARD für "USER" die mehr machen wollen!

Modernes Keyboard eines fuhrenden deutschen Markenherstellers, sehr fläche Bauform, mit eingesetzten Blindstucken, so daß der Tastenausschnitt eine Rechteckform ergibt, insges. 80 Tasten mit sep. abgesetztem 10er-Block sowie zusätzlichem Funktionestastenblock, Tastenkontakte – **Hallgeneratoren** — zusätzlich 8 LED für Progr.-Anzeige und einige Tasten mit LED-Beleuchtung. Das Interface ist mit **Z-80-CPU/PIO/PIO/SIO/CTC** bestuckt, die Tasten des 10er-Blocks und des Funktionsblock (19 Tasten) sind mit — **STRINGS** — belegbar. Die Umschaltung dieser 17 Tasten sind mit jeweils 3 Zeichenfolgen vorbelegt, die 4 verschiedenen Speicherebenen werden jeweils mit LEDs angezeigt, zwei Tasten dienen als Vortaste.
In den Funktions-Ebenen sind die vorbelegten — STRINGS — (EPROM) jederzeit anderbar (RAM). Stringkapazität: 19 Strings i.d. Grundebene und 4×3 Strings un den Funktionsebenen, alle im Betrieb änderbar (Stringlänge max. 250 Zeichen).

Techn. Daten: Alpha-Numerische Tastatur **mit ASCII-Parallel-Norm**, serielle-Daten-Übernahme vorgesehen, deutscher Zeichensatz oder Umschaltung auf US-Zeichensatz, Span.-Versorgung: +5 V-DC, Datenübernahme jeweils im 16-Pin-DIL, wobei **ASCII-Norm** fur APPLE-Anwender belegt ist.
Betriebsbereites Keyboard mit Interface und allen Schaltunteragen sowie EPROM-Listing vom Keyboard und Interface sind beigefügt. **DM 425,–**
Prospekt mit allen Schaltunterlagen gegen Schutzgebühr von **DM 12,–**
Passendes Keyboard-Gehäuse in ABS-Kunststoff, Farbe APPLE-beige mit braunem Unterteil **DM 112,–**

**Spezial KEYBOARD-Spiralkabel**
Ideale Verbindung zwischen Motherboard und Keyboard, hochflexibel, kapazitätsarm, mit 16 PIN-DIL-Stecker bis 1,5 m ausziehbar, Farbe: Schwarz ..... **DM 29,50**

Setzen SIE Ihren Fernseher als MONITOR ein.
Einfach mit Hilfe des HF-Modulators UM 1111 E 36. Dieses Modul setzt beliebige oder digitale Signale auf UHF-Eingänge z.B. K36 um. Techn. Daten: Frequenzbereich 50–800 MHz, Bandbreite 4–8 MHz; HF-dichtes Metallgehäuse, Maße 69 x 35 x 18 mm. U = 3,5–12 V. ..... **DM 19.–**

**GALVANO-Goldstift**
Galvanisieren Sie jetzt in wenigen Minuten blanke metallene Gegenstände mit einer richtigen GOLDSCHICHT. Daher ideal zum Vergolden von Leiterbahnen, Kontaktpunkte auf Platinen, beschädigte Double-Oberflächen usw. ..... **DM 29.50**
Spez.-Batterie 5,6 V/27 PX ..... **DM 9.90**

**ITT – Querstromlüfter – OLZ 06/0018 A,**
jedoch mit links angeschlagenem Motor für wahlweise waagerechte oder senkrechte Betriebslage, mit guter und regelbarer Förderleistung 220 V/50 Hz. 1500 U/min. Maße: 253 x 110 x 90 mm ..... **DM 29,50**

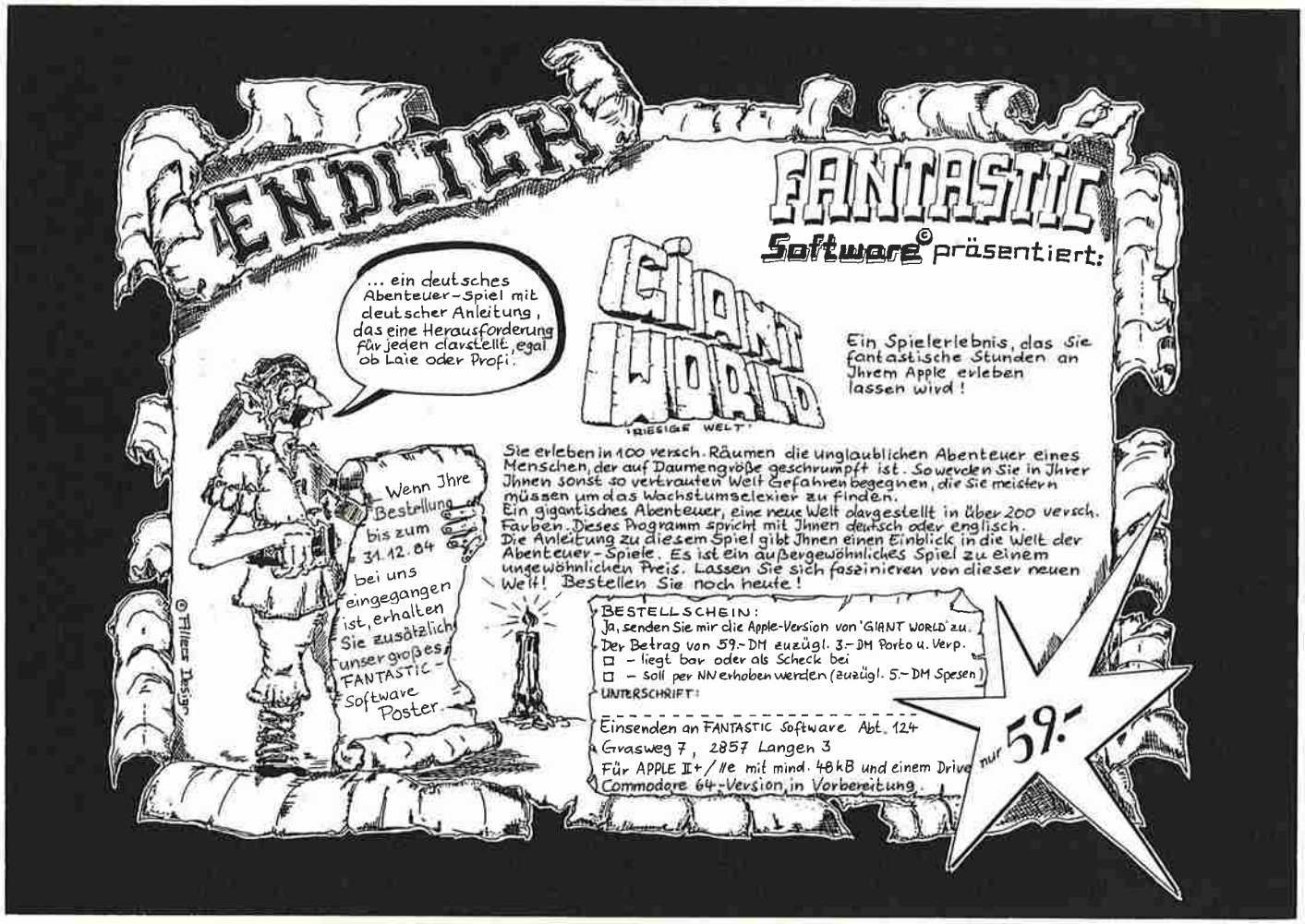
**AXIAL-Lüfter, Typ TA 450 S**
Kompakt-Lüfter für die Kühlung von Netzteilen, Leistungsendstufen sowie andere Kühlzwecke. Schwarzes Alu-Gehäuse, Maße: 120 x 120 x 38 mm; 5 Turbinenflügel, solide Industrie-Ausführung für Dauerbetrieb, US-Fab. – TORIN – Anschluß 220 V–50 Hz. Spaltpolmotor 14 W, 2500 UpM, sehr leise, Förderleistung ca. 110 cbm/h **DM 35,–**

Verkaufsbedingungen: Versand ab DM 20,–; Porto und Verpackung per Nachnahme oder Vorauskasse + Porto; alle Preis inkl. Mehrwertsteuer

**NADLER electronic**
**4000 DÜSSELDORF**
**Telefon (0211) 35 04 49**
**Kurfürstenstraße 39**

```
6085: AD 0B 60   87            LDA  XACHSEH
6088: 8D 41 62   88            STA  DIVDENDH
608B: 20 3D 62   89            JSR  DIVO
                 90   *
608E: AE 46 62   91            LDX  DIVRESTL
6091: BD 2E 62   92            LDA  BITTABLE,X
6094: 8D 3C 62   93            STA  BITBYTE
                 94   *
6097: AD 44 62   95            LDA  DIVQUOTL
609A: 4A         96            LSR
609B: 90 05      97            BCC  STAPG2
609D: AC 54 C0   98   STAPG1   LDY  $C054       ;PAGE1
60A0: B0 03      99            BCS  PAGER
60A2: AC 55 C0   100  STAPG2   LDY  $C055       ;PAGE2
                 101  *
60A5: A8         102  PAGER    TAY
60A6: B1 CE      103           LDA  (IND1),Y
60A8: 0D 3C 62   104           ORA  BITBYTE
60AB: 91 CE      105           STA  (IND1),Y
60AD: 60         106           RTS
                 107  *
                 108  * 192 ZEILEN 0-191: Y-ACHSE
                 109  * -------------------------
                 110  *
60AE: 00 00 00   111  HGRADRL  HEX  0000000000000000
60B1: 00 00 00 00 00
60B6: 80 80 80   112           HEX  8080808080808080
60B9: 80 80 80 80 80
60BE: 00 00 00   113           HEX  0000000000000000
60C1: 00 00 00 00 00
60C6: 80 80 80   114           HEX  8080808080808080
60C9: 80 80 80 80 80
60CE: 00 00 00   115           HEX  0000000000000000
60D1: 00 00 00 00 00
60D6: 80 80 80   116           HEX  8080808080808080
60D9: 80 80 80 80 80
60DE: 00 00 00   117           HEX  0000000000000000
60E1: 00 00 00 00 00
60E6: 80 80 80   118           HEX  8080808080808080
60E9: 80 80 80 80 80
                 119  *
60EE: 28 28 28   120           HEX  2828282828282828
60F1: 28 28 28 28 28
60F6: A8 A8 A8   121           HEX  A8A8A8A8A8A8A8A8
60F9: A8 A8 A8 A8 A8
60FE: 28 28 28   122           HEX  2828282828282828
6101: 28 28 28 28 28
6106: A8 A8 A8   123           HEX  A8A8A8A8A8A8A8A8
6109: A8 A8 A8 A8 A8
610E: 28 28 28   124           HEX  2828282828282828
6111: 28 28 28 28 28
6116: A8 A8 A8   125           HEX  A8A8A8A8A8A8A8A8
6119: A8 A8 A8 A8 A8
611E: 28 28 28   126           HEX  2828282828282828
6121: 28 28 28 28 28
6126: A8 A8 A8   127           HEX  A8A8A8A8A8A8A8A8
6129: A8 A8 A8 A8 A8
                 128  *
612E: 50 50 50   129           HEX  5050505050505050
6131: 50 50 50 50 50
6136: D0 D0 D0   130           HEX  D0D0D0D0D0D0D0D0
6139: D0 D0 D0 D0 D0
613E: 50 50 50   131           HEX  5050505050505050
6141: 50 50 50 50 50
6146: D0 D0 D0   132           HEX  D0D0D0D0D0D0D0D0
6149: D0 D0 D0 D0 D0
614E: 50 50 50   133           HEX  5050505050505050
6151: 50 50 50 50 50
6156: D0 D0 D0   134           HEX  D0D0D0D0D0D0D0D0
6159: D0 D0 D0 D0 D0
615E: 50 50 50   135           HEX  5050505050505050
6161: 50 50 50 50 50
6166: D0 D0 D0   136           HEX  D0D0D0D0D0D0D0D0
6169: D0 D0 D0 D0 D0
                 137  *
616E: 20 24 28   138  HGRADRH  HEX  2024282C3034383C
6171: 2C 30 34 38 3C
6176: 20 24 28   139           HEX  2024282C3034383C
6179: 2C 30 34 38 3C
617E: 21 25 29   140           HEX  2125292D3135393D
6181: 2D 31 35 39 3D
6186: 21 25 29   141           HEX  2125292D3135393D
6189: 2D 31 35 39 3D
618E: 22 26 2A   142           HEX  22262A2E32363A3E
6191: 2E 32 36 3A 3E
6196: 22 26 2A   143           HEX  22262A2E32363A3E
6199: 2E 32 36 3A 3E
```

```
619E: 23 27 2B   144           HEX  23272B2F33373B3F
61A1: 2F 33 37 3B 3F
61A6: 23 27 2B   145           HEX  23272B2F33373B3F
61A9: 2F 33 37 3B 3F
                 146  *
61AE: 20 24 28   147           HEX  2024282C3034383C
61B1: 2C 30 34 38 3C
61B6: 20 24 28   148           HEX  2024282C3034383C
61B9: 2C 30 34 38 3C
61BE: 21 25 29   149           HEX  2125292D3135393D
61C1: 2D 31 35 39 3D
61C6: 21 25 29   150           HEX  2125292D3135393D
61C9: 2D 31 35 39 3D
61CE: 22 26 2A   151           HEX  22262A2E32363A3E
61D1: 2E 32 36 3A 3E
61D6: 22 26 2A   152           HEX  22262A2E32363A3E
61D9: 2E 32 36 3A 3E
61DE: 23 27 2B   153           HEX  23272B2F33373B3F
61E1: 2F 33 37 3B 3F
61E6: 23 27 2B   154           HEX  23272B2F33373B3F
61E9: 2F 33 37 3B 3F
                 155  *
61EE: 20 24 28   156           HEX  2024282C3034383C
61F1: 2C 30 34 38 3C
61F6: 20 24 28   157           HEX  2024282C3034383C
61F9: 2C 30 34 38 3C
61FE: 21 25 29   158           HEX  2125292D3135393D
6201: 2D 31 35 39 3D
6206: 21 25 29   159           HEX  2125292D3135393D
6209: 2D 31 35 39 3D
620E: 22 26 2A   160           HEX  22262A2E32363A3E
6211: 2E 32 36 3A 3E
6216: 22 26 2A   161           HEX  22262A2E32363A3E
6219: 2E 32 36 3A 3E
621E: 23 27 2B   162           HEX  23272B2F33373B3F
6221: 2F 33 37 3B 3F
6226: 23 27 2B   163           HEX  23272B2F33373B3F
6229: 2F 33 37 3B 3F
                 164  *
                 165  * Bit-Tabelle
                 166  * -----------
                 167  *
622E: 01         168  BITTABLE DFB  %00000001  ;0
622F: 02         169           DFB  %00000010  ;1
6230: 04         170           DFB  %00000100  ;2
6231: 08         171           DFB  %00001000  ;3
6232: 10         172           DFB  %00010000  ;4
6233: 20         173           DFB  %00100000  ;5
6234: 40         174           DFB  %01000000  ;6
                 175  *
6235: 01         176           DFB  %00000001  ;0
6236: 02         177           DFB  %00000010  ;1
6237: 04         178           DFB  %00000100  ;2
6238: 08         179           DFB  %00001000  ;3
6239: 10         180           DFB  %00010000  ;4
623A: 20         181           DFB  %00100000  ;5
623B: 40         182           DFB  %01000000  ;6
                 183  *
623C: 00         184  BITBYTE  HEX  00
                 185  *
                 186  *
                 187  * Division 16 Bit : 16 Bit
                 188  * ------------------------
                 189  *
623D: 4C 4B 62   190  DIVO     JMP  DIV1
                 191  *
                 192  * DIVIDEND + DIVISOR
                 193  *
6240: 00         194  DIVDENDL HEX  00
6241: 00         195  DIVDENDH HEX  00
6242: 07         196  DIVISORL HEX  07
6243: 00         197  DIVISORH HEX  00
                 198  *
                 199  * QUOTIENT + DIVISIONSREST
                 200  *
6244: 00         201  DIVQUOTL HEX  00
6245: 00         202  DIVQUOTH HEX  00
6246: 00         203  DIVRESTL HEX  00
6247: 00         204  DIVRESTH HEX  00
                 205  *
6248: 00         206  AREG     HEX  00
6249: 00         207  XREG     HEX  00
624A: 00         208  YREG     HEX  00
                 209  *
624B: 8D 48 62   210  DIV1     STA  AREG
624E: 8E 49 62   211           STX  XREG
6251: 8C 4A 62   212           STY  YREG
                 213  *
```




## Nutzen Sie Ihren Apple-Computer auch zum Lernen

INTUS-Lernprogramme sind attraktiv und motivierend für vergnügliches Lernen mit hohem Lernerfolg.
Alle Programme in deutscher Sprache für Apple IIe und IIc.

- Maschineschreiben wie der Blitz. Didaktisch ausgezeichnet. Garantiert erfolgreich in 20 Lektionen. DM 188,--
- Basic-Lernprogramm. Bestes Lernprogramm 1982 in USA. DM 295,--
- Computer-Simulator. Mit Lehrgang in 5 Lektionen. DM 146,--
- LOGO deutsch. "Das neue Lernen" – Computersprache für Anfänger und "Profis". (Probedisk DM 25,--) DM 395,--
- Deutsche Grammatik mit Spaß. Für Sekretärinnen, Schüler und "Jedermann". Einschl. Rechtschreibe-Training mit Vortest/Nachtest. 12 Bereiche, je Disk DM 165,--
- Wortschatztrainer, Franz./Deutsch, 800 Wörter DM 140,--
  Englisch/Deutsch, 800 Wörter DM 140,--
  Business English, 1200 Wörter DM 163,--
  Attention please, 2400 Wörter DM 280,--
- An der Börse jonglieren. Praxisgerechtes, aufregendes, lehrreiches Börsengeschehen. DM 107,--
- Unternehmens-Planspiel "Kartellbrüder und Halsabschneider". Professionelles, lehrreiches Strategiespiel. Vorkenntnisse notwendig. Umfangreiche Dok. DM 495,--
- Schulprogramme wie Rechnen/Mathe, Sprachen, Physik, Chemie, Informatik, Vorschule. Bestens für die Nachhilfe geeignet. Vermietung von Programmen.

★★★★ Mit Ihrer ersten Bestellung erhalten Sie gratis das Programm "Mein Freund der Apple" mit interessanten Denk-Strategie- und Entspannungsspielen.

INTUS LERN-SYSTEME AG
Kaiserstraße 21 · 7890 Waldshut-Tiengen · Telefon 07751-7920

*Programmierzeit auf die Hälfte gekürzt!*
*Speicherbedarf auf ein Drittel reduziert!*

# EXBASIC® LEVEL II™

*Mit EXBASIC LEVEL II können Sie Ihren Computer um ein Vielfaches einfacher und effizienter programmieren. Sie haben über 75 neue, äußerst leistungsfähige Funktionen, die den Erfordernissen moderner Software-Erstellung entsprechen. Für Computer von ...*

apple® commodore®

*EXBASIC LEVEL II ist bereits weltweit im Einsatz. Schreiben Sie sofort „Schicken Sie kostenlos ausführliche Informationen zu EXBASIC LEVEL II für..... (genauen Computertyp angeben!)"*

INTERFACE AGE
Verlag GmbH
Josefsburgstraße 6
8000 München 80
Tel.: 089/434089

```
6254: A9 00      214          LDA  #0
6256: 8D 46 62   215          STA  DIVRESTL
6259: 8D 47 62   216          STA  DIVRESTH
                 217  *
625C: AD 42 62   218          LDA  DIVISORL
625F: D0 0D      219          BNE  DIV2
6261: AD 43 62   220          LDA  DIVISORH
6264: D0 08      221          BNE  DIV2
                 222  *
                 223  * DIVISION DURCH NULL VERBOTEN
                 224  *
6266: 8D 44 62   225          STA  DIVQUOTL
6269: 8D 45 62   226          STA  DIVQUOTH
626C: F0 36      227          BEQ  DIV5
                 228  *
626E: AD 40 62   229  DIV2    LDA  DIVDENDL
6271: 8D 44 62   230          STA  DIVQUOTL
6274: AD 41 62   231          LDA  DIVDENDH
6277: 8D 45 62   232          STA  DIVQUOTH
                 233  *
627A: A0 10      234          LDY  #$10        ;16BITS
627C: 0E 44 62   235  DIV3    ASL  DIVQUOTL
627F: 2E 45 62   236          ROL  DIVQUOTH
6282: 2E 46 62   237          ROL  DIVRESTL
6285: 2E 47 62   238          ROL  DIVRESTH
6288: 38         239          SEC
6289: AD 46 62   240          LDA  DIVRESTL
628C: ED 42 62   241          SBC  DIVISORL
628F: AA         242          TAX
6290: AD 47 62   243          LDA  DIVRESTH
6293: ED 43 62   244          SBC  DIVISORH
6296: 90 09      245          BCC  DIV4
6298: 8E 46 62   246          STX  DIVRESTL
629B: 8D 47 62   247          STA  DIVRESTH
629E: EE 44 62   248          INC  DIVQUOTL
                 249  *
62A1: 88         250  DIV4    DEY
62A2: D0 D8      251          BNE  DIV3
                 252  *
62A4: AD 48 62   253  DIV5    LDA  AREG
62A7: AE 49 62   254          LDX  XREG
62AA: AC 4A 62   255          LDY  YREG
62AD: 60         256          RTS

686 bytes

DHGR.LINEPLOTTER

100 REM Double-Hires-Lineplotter
110 PRINT  CHR$ (21): REM 80-Zeichenkarte abstellen
120 CE = 49165: REM 80COL ein
130 CA = 49164: REM 80COL aus
140 HE = 49239: REM HIRES ein
150 HA = 49238: REM HIRES aus
160 TE = 49233: REM TEXT ein
170 TA = 49232: REM TEXT aus
180 ME = 49235: REM MIXED ein
190 MA = 49234: REM MIXED aus
```

```
200 SE = 49153: REM 80STORE ein
210 SA = 49152: REM 80STORE aus
220 PE = 49237: REM PAGE2 ein
230 PA = 49236: REM PAGE2 aus
240 AE = 49247: REM AN3 ein
250 AA = 49246: REM AN3 aus
300 REM Double Hires initialisieren
305 HGR : REM untere Seite löschen
310 P =  PEEK (MA): REM Mixed aus
312 POKE SE,0: REM 80-Store ein
314 POKE CE,0: REM 80-Column ein
316 P =  PEEK (PE): REM Page 2 (obere  Seite)
350 CALL 62450: REM untere Seite löschen
360 P =  PEEK (AA): REM AN 3 aus (Double Hires)
400 REM Lineplotter
405 HCOLOR= 3
410 X1 = 100:Y1 = 50:X2 = 200:Y2 = 50
420 GOSUB 500
430 X1 = 200:Y1 = 50:X2 = 200:Y2 = 150
440 GOSUB 500
450 X1 = 200:Y1 = 150:X2 = 100:Y2 = Y1
460 GOSUB 500
470 X1 = 100:Y1 = 150:X2 = X1:Y2 = 50
480 GOSUB 500
490 X1 = 100:Y1 = 50:X2 = 200:Y2 = 150
495 GOSUB 500
498 GOTO 805
500 XD = X2 - X1:YD = Y2 - Y1
510 XS = 1:YS = 1
520 IF XD < 0 THEN XS =  - 1:XD =  ABS (XD)
530 IF YD < 0 THEN YS =  - 1:YD =  ABS (YD)
540 SUM = 0: IF XD = 0 THEN SUM =  - 1
550 I = X1: GOSUB 900
560 HPLOT XX,Y1
570 IF X1 = X2 THEN  IF Y1 = Y2 THEN  RETURN
580 IF SUM < 0 THEN Y1 = Y1 + YS:SUM = SUM + XD: GOTO 550
590 X1 = X1 + XS:SUM = SUM - YD: GOTO 550
800 REM Double Hires abschalten
805 GET A$
810 P =  PEEK (PA): REM untere Seite einschalten
820 P =  PEEK (TE): REM Text einschalten
830 P =  PEEK (AE): REM Double Hires ausschalten
840 POKE CA,0: REM 40-Column
850 POKE SA,0: REM 40-Store
860 TEXT
870 END
890 REM X-Koordinate berechnen und auf richtige Seite
    schalten
900 U = I / 7:BYTE =  INT (U):Z = BYTE / 2:PG = Z -  INT
    (Z):XX =  INT (( INT (Z) + U - BYTE) * 7 + .5)
910 REM Seite wählen
920 P =  PEEK (PE): REM obere Seite einschalten
930 IF PG THEN P =  PEEK (PA): REM untere Seite
    einschalten
940 RETURN
M untere Seite
    einschalten
940 RETURN
```


**Neu im Hüthig Software Service**

# SUPERPLOT

**Double-Hires-Utility**

von Karl-Walter Bott, erscheint Mitte Dezember 1984, Programmdiskette und Manual, DM 48,–

SUPERPLOT ist eine neue, ungewöhnlich kompakte und schnelle Ampersand-Utility für Double Hires, die einschließlich eines vollständigen ASCII-Shape-Zeichensatzes wahlweise in Bank 1 oder Bank 2 der Language Card liegt und damit sowohl unter ProDOS als auch unter DOS 3.3, falls letzteres in die LC-Bank geschoben wurde, benutzt und in eigene Applesoftprogramme integriert werden kann. SUPERPLOT unterstützt die üblichen HGR-Befehle, denen lediglich ein & vorangestellt werden muß, also z. B. & HPLOT 500, 100 TO 500, 150 usw. SUPERPLOT ist speziell für das Plotten von beschrifteten wissenschaftlichen Funktionskurven mit hoher Auflösung gedacht und weniger für HGR-Spiele.

**Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1**

# apple

**Aktion Weihnachts-preise:**

## Macintosh

**128 KB incl. MacWrite/Paint**

**5.875,–**

| | |
|---|---|
| Externes Macintosh-Laufwerk 400 KB | 1.245,– |
| Imagewriter | 1.490,– |

## Apple *IIc*

**128 KB**

**2.999,–**

| | |
|---|---|
| Original //c-Monitor mit Stand | 594,– |
| Externes Laufwerk //c | 895,– |

## Apple *IIe*

**64 KB, Disk. m. Controller und original *IIe*-Monitor**

**3.350,–**

| | |
|---|---|
| Apple //e-Duodisk-Paket: //e 64 KB, Apple 80Z/64K, Duodisk-Laufwerk, original //e-Monitor | 4.590,– |
| Duodisk-Laufwerk kpl. m. Controller | 1.957,– |
| Disk // mit Controller | 995,– |
| Disk // ohne Controller | 795,– |
| Apple 80Z/64K-Karte | 689,– |

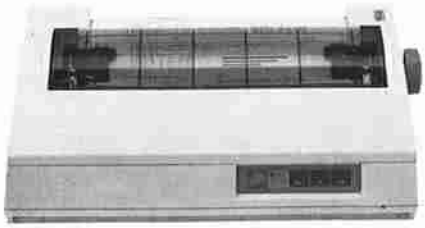
**Wichtig :** Alle Preise incl. MwSt. Versandkosten DM 8,– (Ausland DM 25,–). Lieferung per Nachnahme oder auf Vorkasse-Scheck. Preiskorrekturen bei Änderung der Herstellerkonditionen (z. B. Dollarkurs-Anpassungen) vorbehalten. Lieferverzögerungen aufgrund von Engpässen beim Hersteller sind möglich!

Jetzt bestellen bei:

**Handelsgemeinschaft I. Müller**

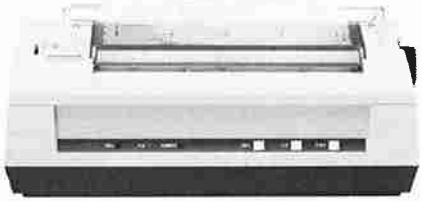
**Sudetenweg 88 · 2150 Buxtehude 3**

# Drucker von C. Itoh für Ihren Apple



**NEU: Modell 7500AP**
- Druckgeschwindigkeit 105 Zeichen/sec.
- 2 kByte Bufferspeicher
- graphikfähig
- Endlospapier- und Einzelblattverarbeitung

nur: **DM 1.179,––** inkl. MwSt.



**8510BP**
- Druckgeschwindigkeit 120 Zeichen/sec.
- 2 kByte Bufferspeicher
- freiladbarer Zeichensatz (96 ASCII-Zeichen)
- 41 nationale Sonderzeichen 64 graphische
- 64 griechische Sonderzeichen
- Bidirektionaldruck durch integrierten Traktor und Friktionswalze

nur: **DM 1.419,––** inkl. MwSt.

**8510SCP**
- Matrixdrucker mit 7 Farben
- Druckgeschwindigkeit 180 Zeichen/sec.
- 2 kByte Bufferspeicher
- erweiterte Darstellungsmöglichkeiten
- Möglichkeit zur farbigen Hardcopy der Apple Graphik

nur: **DM 2.399,––** inkl. MwSt.

**8510SP**
- gleiche technische Merkmale wie 8510SCP jedoch keine Farbdarstellung möglich

nur: **DM 1.798,––** inkl. MwSt.

## Interfacekarten

zum Anschluß an Apple II/IIe und kompatible

**Super-Graphik-Interface**
ermöglicht Anschluß aller parallelen Itoh-Drucker
- mit Interface-Handbuch (32 Seiten)
- 1,5 m Flachleitung mit Steckern anschlußfertig
- nutzt alle Graphikmöglichkeiten des Apple

nur: **DM 199,––** inkl. MwSt.

**Farb-Graphik-Interface für 8510SCP**
ermöglicht die farbige Hardcopy der Apple HIRES-Seiten
- mit Interface-Handbuch
- Leitung mit Steckverbindungen anschlußfertig für die Itoh Color-Geräte

nur: **DM 282,––** inkl. MwSt.

**Elektronik-Vertrieb Köller**

COMPUTER UND ZUBEHÖR
Lothe-Niesetal-Straße 4
4938 Schieder-Schwalenberg
Tel. (05233) 7550

**r+r electronic**
Elektronik · Fachliteratur · Personal-Computer

## Literatur für APPLE II, IIe, IIc und kompatible Geräte

### APPLESOFTBASIC

**Apple II Anwenderhandbuch** 56,00
von Lon Poole, 1982, 450 Seiten, te-wi Verlag
Diese Buch erspart Ihnen zeitraubendes und nutzloses Suchen nach der wirklich verwendbaren Dokumentation für Ihren Apple. Neben der ausführlichen Auskunft über Peripherie, Zubehör und Drucker wird gezeigt, wie man in BASIC und in Maschinensprache programmiert.

**BASIC-Wegweiser** für den Apple II 32,–
von Ekkehard Kaier, 1984, 200 Seiten, Vieweg-Verlag
Diese Buch weist Wege zum erfolgreichen Einsatz von Computern der Apple II-Familie mit Applesoft-BASIC. Mit über 80 Programmen, 7 Dateien, 24 PAPs und 84 Bildern.

**APPLE II Leicht gemacht** 28,00
von J. Kasomer, 1984, 192 Seiten, SYBEX-Verlag
Dieses Buch zeigt wie Sie Ihren APPLE IIe, II+ oder IIc in wenigen Stunden voll einsetzen können. Sie lernen wie leicht es ist Ihr eigenes BASIC-Programm zu schreiben.

**BASIC Übungen für den APPLE** 38,00
von J.-P. Lamoitier, 1983, 256 Seiten, SYBEX-Verlag
Das Buch enthält eine Reihe von abgestuften Übungen mit zunehmendem Schwierigkeitsgrad. So werden Ihre Programmierfähigkeiten aufgebaut und an vielen Beispielen erprobt.

**APPLE II BASIC Handbuch** 32,00
von D. Hergert, 1984, 304 Seiten, SYBEX-Verlag
Ein handliches Nachschlagwerk für Ihren APPLE II, IIe und IIc Tips und Vorschläge machen das Programmieren einfacher u. effizienter. Das Buch für Anfänger und Fortgeschrittene.

**Spiele für den APPLE** 38,00
von M.J. Capella/M.D. Weinstock, 270 Seiten, 1984, M&T-Buchverlag
Eine Sammlung von bewährten alten und raffinierten neuen Spielen für Ihren APPLE-Computer.

**Spielprogramme für den Applelle** 32.–
von H. Franklin/J. Koltnow/L. Finkel, 1984, Vieweg-Verlag
Das Buch stellt Spiele vor und zeigt, wie man Spiele erfindet.

### PASCAL

**Apple II Pascal** 56,00
von A. Lührmann/H. Peckham, 1982, te-wi Verlag
Sie benötigen zu diesem Buch keine Vorkenntnisse, sondern lernen an Hand von Beispielen und Übungen, wie man selber PASCAL-Programme entwickelt und sie austestet.

### Diskettenbetriebssysteme

**APPLE DOS 3.3** - Tips und Tricks 28,00
von U. Stiehl, 1984, 216 Seiten, Hüthig-Verlag
Dies ist die erste deutschsprachige Darstellung des Betriebssystems DOS 3.3 für den APPLE II/II Plus/IIe.
**Begleitdiskette zu APPLE DOS 3.3** ..... 28,00

**APPLE ProDOS** für Aufsteiger Bd. I 28,00
von U. Stiehl, 1984, 203 Seiten, Hüthig-Verlag
Nachfolgeband zu APPLE DOS 3.3 für das neue Betriebssystem ProDOS. Assembler Programmierer finden in diesem Band eine wertvolle Unterstützung durch interne ProDOS Systemadressen oder interessanten Programmen.
**Begleitdiskette zu APPLE ProDOS Band I** ..... 28,00

### Assembler

**APPLE Assembler** - Tips und Tricks 34,00
von U. Stiehl, 1984, 227 Seiten, Hüthig Verlag
Das neue jetzt vorliegende Buch enthält alle Monitor und Applesoft Interpreter Routinen für All/IIe/IIc (40 Z./Z.) wie eine Einführung in die 6502 Assembler Programmierung.
**Begleitdiskette zu APPLE Assembler** ..... 28,00

**Programmierung des 6502** 44,00
von Rodnay Zaks, 1984, 288 Seiten, SYBEX-Verlag
Sehr gut verständliche Einführung in die Assembler-Programmierung mit dem Mikroprozessor 6502. Bestens geeignet für Anfänger und Fortgeschrittene.

**6502 Anwendungen** 38,00
von Rodnay Zaks, 1983, 288 Seiten, SYBEX-Verlag
Das Eingabe/Ausgabe-Buch für Ihren 6502 Mikroprozessor. Viele Anwendungsbeispiele helfen Ihnen, das Erlernte in die Praxis umzusetzen.

**Fortgeschrittene 6502 Programmierung** 42,00
von Rodnay Zaks, 1984, 288 Seiten, SYBEX-Verlag
Lernen Sie wie man schwierige Probleme mit dem 6502 löst. Ein muß für jeden fortgeschrittenen Apple-User.

**6502/65C02 Maschinensprache** 49,00
von C. Persson, 1983, 250 Seiten, Verlag Heinz Heise
Diese Buch eines deutschen Autors ist eine intensive, praxisgerechte Einführung in die Programmierung des 6502. Als erstes Buch auf dem dt. Markt behandelt es auch den 65C02.

**APPLE Maschinensprache** 49,00
von D. Inman/K. Inman, 300 Seiten, 1984, te-wi Verlag
Eine Schritt-für-Schritt-Einweisung in die professionellere Entwicklung von Maschinenprogrammen mit APPLE SYSTEM MONITOR und APPLE MINI-ASSEMBLER.

**Alle Bücher sofort ab Lager lieferbar. Bestellen Sie per NN oder besuchen Sie uns. Wir führen eine große Auswahl an aktuellen Büchern und Zeitschriften.**
6900 Heidelberg 1 Breslauerstr. 29 Tel. 06221/781500
**Geschäftszeiten:** Mo. - Fr. 9-13 +14 - 18 Sa. 9 -13 Uhr
**Versandanschrift:** 6900 Heidelberg 1 Dammweg 2

# Modems kom

## Praktische Ratschläge für die Telekommunikatio

von Matthias Pohl

**Datenaustausch und Kommunikation mittels Modem stellt sicher eine der reizvollsten Anwendungen für die mittlerweile weit verbreiteten Home- und Personal-Computer dar. Auch in Deutschland vergrößert sich die Schar der Modembesitzer von Tag zu Tag. Erste Informationen über das Was und Wie will nachfolgender Artikel geben.**

Um die Kommunikation oder Datenübertragung zwischen zwei voneinander weit entfernt stehenden Rechnern zu ermöglichen, bedarf es der Zwischenschaltung einer Datenübertragungseinheit, eines Modems. Das Modem verwandelt die vom Computer gelieferten elektrischen Signale in Schallwellen, die dann über eine Telefonleitung zu einem anderen Computer an einen beliebigen Punkt der Erde geleitet und dort von einem anderen Modem wieder zurückverwandelt werden. Die Bezeichnung Modem leitet sich aus den Anfangsbuchstaben von MOdulator/DEModulator ab: Die zu übertragenden Informationen werden vom Sender auf eine Trägerfrequenz (2100 Hz) moduliert und vom Empfänger demoduliert. Übrigens sagt man meistens das Modem (Neutrum), des Modems (Gen.), die Modems (Pl.), während im DUDEN noch der Modem (Maskulinum) steht, was allerdings nicht (mehr) üblich ist.

### Die Schnittstelle

Wie sieht nun eine funktionsfähige Konfiguration für den Apple II/IIe/IIc aus und wie gestaltet sich die Bedienung? Neben dem eigentlichen Rechner mit Bildschirm und dem Modem wird eine V24-Schnittstelle (RS 232-C) benötigt, über die die Rechner der Apple-II-Familie (mit Ausnahme des IIc) nicht serienmäßig verfügen. Erst die V24-Schnittstelle ermöglicht aufgrund ihrer Standardisierung die serielle Datenübertragung zwischen Geräten verschiedener Hersteller. Sehr bewährt hat

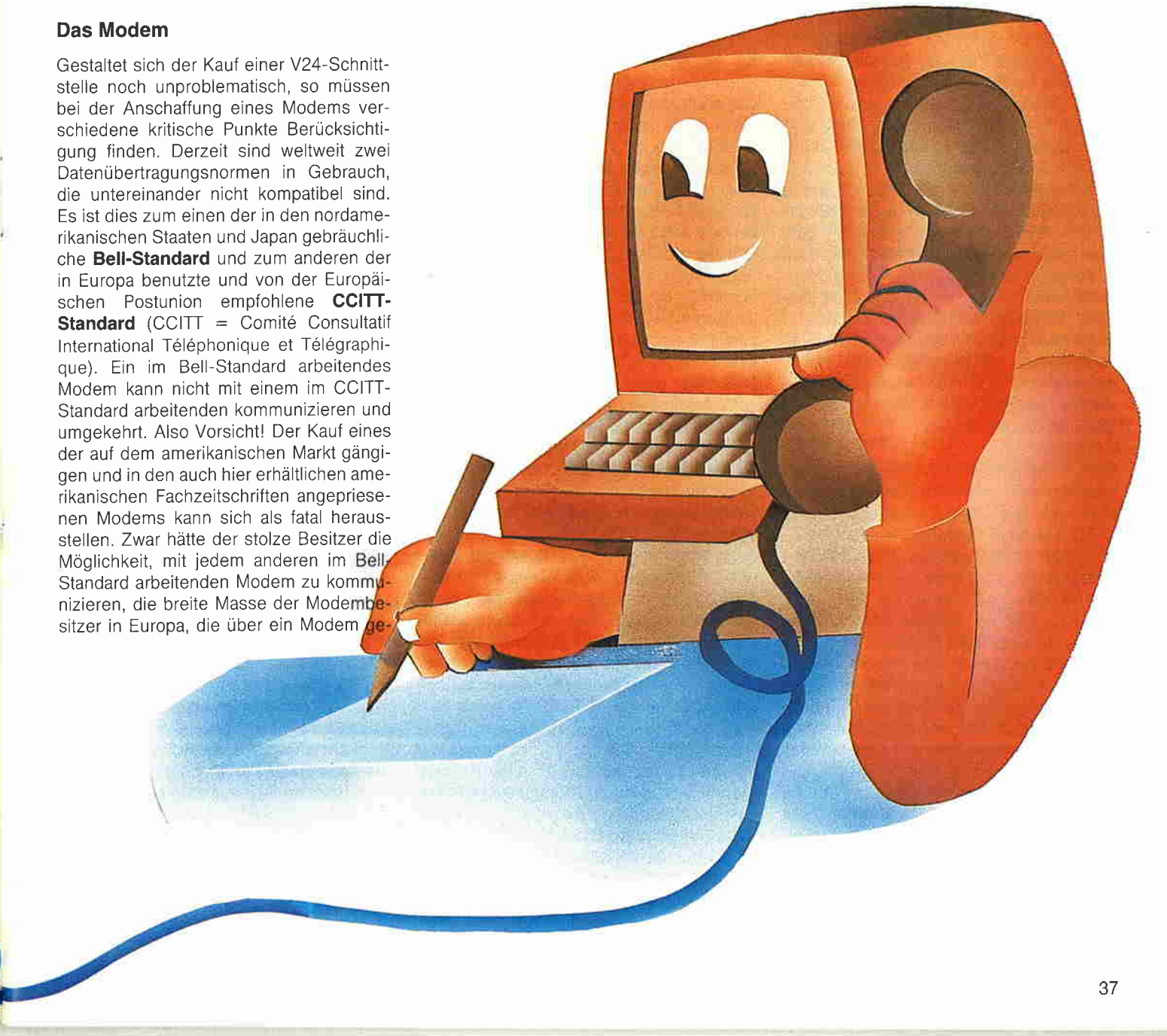
# men in Mode

## mit Modem

sich hier die Super Serial Card der Firma Apple, die über einen speziellen Kommunikationsmodus verfügt. Selbstverständlich kann aber auch jedes andere Interface, das die Besonderheiten des Apple-Rechners berücksichtigt, verwandt werden.

### Das Modem

Gestaltet sich der Kauf einer V24-Schnittstelle noch unproblematisch, so müssen bei der Anschaffung eines Modems verschiedene kritische Punkte Berücksichtigung finden. Derzeit sind weltweit zwei Datenübertragungsnormen in Gebrauch, die untereinander nicht kompatibel sind. Es ist dies zum einen der in den nordamerikanischen Staaten und Japan gebräuchliche **Bell-Standard** und zum anderen der in Europa benutzte und von der Europäischen Postunion empfohlene **CCITT-Standard** (CCITT = Comité Consultatif International Téléphonique et Télégraphique). Ein im Bell-Standard arbeitendes Modem kann nicht mit einem im CCITT-Standard arbeitenden kommunizieren und umgekehrt. Also Vorsicht! Der Kauf eines der auf dem amerikanischen Markt gängigen und in den auch hier erhältlichen amerikanischen Fachzeitschriften angepriesenen Modems kann sich als fatal herausstellen. Zwar hätte der stolze Besitzer die Möglichkeit, mit jedem anderen im Bell-Standard arbeitenden Modem zu kommunizieren, die breite Masse der Modembesitzer in Europa, die über ein Modem ge-



mäß CCITT verfügen, bliebe ihm aber verschlossen. Außerdem bestände keinerlei Möglichkeit zur Benutzung von Datex-P, einer Einrichtung der Deutschen Bundespost (DBP), von der noch zu sprechen sein wird.

Von verschiedenen Händlern werden seit einiger Zeit Modems angeboten, die nachträglich auf CCITT umgerüstet worden sind. Auch hier ist Vorsicht geboten! Es hat sich nämlich wiederholt gezeigt, daß diese zwar im *Originate-Modus* korrekt arbeiten, nicht aber im *Answer-Modus*. Sie sind also imstande, als Anrufer eine Verbindung herzustellen, erkennen es jedoch nicht, wenn sie von einem anderen Rechner angerufen werden.

Fast alle erhältlichen Modems erlauben den Betrieb mit Übertragungsgeschwindigkeiten von 300-1200 Baud (Bit/s). 1 Byte entspricht 7-8 Bits, doch wegen der Stoppbits usw. gilt als Faustregel: 10 Bits = ca. 1 Byte (Zeichen: Buchstaben, Ziffer usw.). Eine Übertragungsrate von 300 Baud bedeutet also ca. 30 Bytes/Sekunde. Unbedingt erforderlich ist die Geschwindigkeitsklasse 300 Baud, da alle Mailboxen in der BRD mit dieser Übertragungsrate arbeiten. Wer sich jetzt zum Kauf eines Modems entschließt, sollte sich überlegen, ob er nicht ein wenig mehr investieren will für ein **BTX-fähiges Modem**. BTX und die in anderen Ländern bereits eingeführten Systeme arbeiten mit 1200/75 Baud. Dies bedeutet, daß mit einer Rate von 1200 Baud Daten empfangen werden, während gleichzeitig mit 75 Baud Daten gesendet werden.

Eine weitere Hürde setzt die DBP. Diese verlangt als Eigner des Leitungsnetzes, über das die Datenübertragung erfolgt, daß alle an das Telefonnetz angeschlossenen elektrischen oder elektroakustischen Zusatzgeräte über eine fernmeldetechnische Zulassungsnummer der DBP (sog. FTZ-Nr., FTZ = Fernmeldetechnisches Zentralamt) verfügen. Diese kann keines der handelsüblichen Modems vorweisen. Man muß dazu anmerken, daß die Bundespost Modems zu horrenden Preisen vermietet. Und wer wird sich denn schon selber Konkurrenz machen! Des weiteren schreibt die Bundespost die Benutzung eines **Akustikkopplers** zwingend vor. Es ist also nicht zulässig, das Modem direkt an der Telefonsteckdose anzuschließen, obwohl diese Anschlußart gegenüber der Verwendung eines Akustikkopplers wesentliche Vorteile bietet: Zum einen entfällt die Anschaffung des DM 300,- bis DM 700,- teuren Akustikkopplers, zum anderen entfallen Störungen durch Nebengeräusche. Die Deutsche Bundespost wird auf nicht korrekt angeschlossene Geräte oder solche, die zum Anschluß nicht zugelassen sind, allerdings nur dann aufmerksam, wenn von diesen Störungen ausgehen, die den ordnungsgemäßen Betrieb des Leitungsnetzes beeinträchtigen.

## Die Verbindungsaufnahme

Hat man sich von all diesen Hürden nicht demoralisieren lassen und den Rechner via V24-Schnittstelle mit dem Modem und dieses wiederum mit der Telefonleitung verbunden, so kann es fast „losgehen". Bevor man versucht, eine Verbindung zu einem anderen Rechner aufzubauen, sollten Datenübertragungsgeschwindigkeit und -format der Gegenstelle bekannt sein. Es ist einleuchtend, daß Daten nicht schneller gesendet werden dürfen, als der Gegenrechner sie aufnehmen kann. Die miteinander kommunizierenden Systeme müssen also im Takt, d.h. mit derselben Übertragungsgeschwindigkeit arbeiten. Zwar ist die Super Serial Card von Apple in der Lage, Daten mit bis zu 9600 Baud zu senden und zu empfangen, jedoch ist keines der marktgängigen Modems für den Apple fähig, mit mehr als 1200 Baud zu arbeiten. Selbst diese verhältnismäßig langsame Übertragungsrate kommt kaum zum Einsatz: alle Mailboxen und Bulletin Boards in der BRD arbeiten mit 300 Baud. Dies liegt daran, daß bei höheren Geschwindigkeiten die Fehlerrate bei der Übertragung stark ansteigt und aufwendige Fehlererkennungs- und -korrekturverfahren notwendig macht. Durch die langsame Übertragung mit 300 Baud erhöhen sich allerdings auch die Telefonkosten erheblich, da sich die Dauer für die Übertragung einer bestimmten Datenmenge erheblich verlängert. Abschließend ist noch das Datenformat festzulegen. An dieser Stelle sei nur das von den allermeisten Mailboxen verwendete **Datenformat** angegeben (Eine ausführlichere Diskussion würde den Rahmen des Artikels sprengen): 7 Datenbits, 1 Stoppbit, keine Parität.

Nach Einstellen des vorgenannten Datenformats und einer Übertragungsrate von 300 Baud ist es nun möglich, die Verbindung aufzubauen. Dies geschieht in der Regel in mehreren Schritten: Der Rechner und das Modem werden aktiviert, wobei sich letzteres allerdings noch im sogenannten *Local-Mode* befinden sollte. In diesem Betriebszustand werden alle vom Computer kommenden Signale bis zum Ausgang des Modems verarbeitet und anschließend durch das Modem zum Computer zurückgeführt. Über den Telefonapparat wird dann die gewünschte Nummer angewählt. Ist der Gegenrechner empfangsbereit, so meldet er sich mit einem durchgehenden Pfeifton. Jetzt verbleiben im Regelfall 30s, um den Telefonhörer auf den Akustikkoppler zu legen und das Modem auf *Online* zu schalten. Dieses gibt dem angerufenen Rechner dann seinerseits seine Betriebsbereitschaft durch einen Signalton bekannt. Der weitere Ablauf hängt von der Art des angerufenen Systems ab. Mitunter wird jetzt die Eingabe einer **User-ID** (Benutzer-Kennwort) und eines **Passwortes** verlangt; oder aber man gelangt direkt in ein Bildschirm-Menü, das die Anwahl verschiedener Funktionen gestattet. Es sei noch darauf hingewiesen, daß sich der Verbindungsaufbau bei dem (verbotenen) direkten Anschluß des Modems an die Telefonleitung (Parallelschaltung) etwas anders gestaltet. Hier kann der Telefonhörer, nachdem die Verbindung zwischen den beiden Rechnern aufgebaut ist, wieder auf die Gabel zurückgelegt werden. Dadurch können keine Nebengeräusche stören, die sonst leicht zum Zusammenbruch der Verbindung führen.

Etwas anders sieht der Ablauf aus, wenn von außen ein Anruf kommt. Nach Abnahme des Telefonhörers wird dieser in den Akustikkoppler gelegt und das Modem auf Online geschaltet. An dieser Stelle ist auf den Unterschied zwischen Originate- und Answer-Modus einzugehen: soll ein Anruf getätigt werden, so ist die Betriebsart **Originate** (hier etwa: aussenden) zu wählen. Soll dagegen ein Anruf entgegengenommen werden, so muß sich das Modem im **Answer**-Modus befinden.

## Nutzungsmöglichkeiten

Eine der interessantesten Möglichkeiten, die sich dem Besitzer eines Modems bieten, ist die Benutzung von **Mailboxen** (Mb) und **Bulletin Boards**, von denen auch in der BRD mittlerweile eine beachtliche Anzahl existiert. Diese werden von Computer Clubs, Firmen oder Privatpersonen betrieben. Die Benutzung ist im allgemeinen kostenlos. Lediglich für die Einrichtung einer privaten Mailbox, vergleichbar etwa einem Postfach, in dem Nachrichten an den Besitzer dieses Postfaches hinterlegt werden können, verlangen einige Mb-Betreiber eine geringe Jahresge-

bühr. Neben diesem nur für den jeweiligen Besitzer zugänglichen Teil verfügen alle Mb über eine mehr oder weniger große öffentliche Mailbox. Auch diese kann man sich wieder als Postfach vorstellen, zu dem aber jeder Anrufer freien Zugang hat. Es findet sich hier alles, angefangen von Kauf/Verkauf-Gesuchen über Tauschangebote bis hin zu Geburtstagsgrüßen. Es ist letztendlich also ein riesiger Kleinanzeigenmarkt. Jeder kann hier sein Anliegen hinterlassen.

Neben diesen beiden Funktionen, persönliche und öffentliche Mailbox, sind fast immer noch eine Reihe weiterer Funktionen vorhanden, die allerdings von System zu System unterschiedlich ausfallen. So sind häufig Dateien mit Telefonnummern anderer Mailboxen vorhanden sowie eine Datei mit DATEX-P-Rufnummern. Tauchen Fragen oder Probleme zur Benutzung und Bedienung auf, so besteht die Möglichkeit, mit dem SysOp (System Operator, „Systemaffe") Kontakt aufzunehmen.

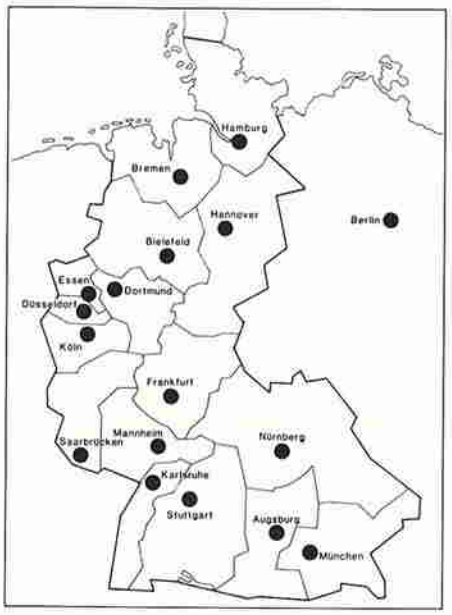
Sehr unterschiedlich fallen bei den verschiedenen Mb die Betriebszeiten aus, in denen sie aktiv sind und angerufen werden können. Eine große Anzahl ist rund um die Uhr erreichbar; andere hingegen sind nur während der Abend- und Nachtstunden aktiv. Auch ist man nicht bei allen Mb's sofort mit dem Computer verbunden, bei manchen meldet sich zuerst der SysOp.

Die Verteilung der in der BRD installierten Mb's schwankt regional sehr stark: Die meisten der 30–40 Mb's befinden sich im Ruhrgebiet, gefolgt vom Großraum Frankfurt–Darmstadt und der Hansestadt Hamburg. Durch keine einzige Mb abgedeckt ist das Ballungsgebiet Rhein-Neckar.

## Die Kosten

Wieviel kostet nun der Spaß? Da die Benutzung der Mb's in allen Fällen kostenlos ist, bleiben nur die üblichen Telefongebühren. Es ist einsichtig, daß fast alle Benutzer auf die tariflich günstigere Zeit von 18.00–8.00h ausweichen. Damit ergibt sich im ungünstigsten Fall eine Gebühr von DM 21.50/h. Nach 18.00h ist aber das „Durchkommen" ein reiner Glücksfall. In den weitaus meisten Fällen ist statt des Signaltones des angerufenen Rechners nur das Besetztzeichen zu hören. Die Lage entspannt sich auch nicht merklich vor Mitternacht, so daß jedem Modembesitzer empfohlen sei, sich mit den „Mächten der Finsternis" auf guten Fuß zu stellen.

Am Rande sei noch vermerkt, daß dieser starke Andrang die meisten Mb-Betreiber veranlaßt hat, ein **Timeout** zu implementieren. Entweder wird die Verbindung nach Ablauf einer bestimmten Zeit (15–30 Minuten) automatisch getrennt und die Leitung für den nächsten Anrufer freigegeben, oder aber die Verbindung wird getrennt, wenn der Mb-Rechner länger als 30s auf eine geforderte Eingabe warten muß. Beide Maßnahmen sollen sicherstellen, daß die Box nicht von einem oder einigen wenigen „Stammkunden" blokkiert wird.




## DATEX-P

Schon mehrfach ist im Vorangegangenen das Stichwort **DATEX-P** (P steht für Daten-**P**aket) gefallen. Es handelt sich hierbei um einen der Datenübertragungsdienste der DBP. Welche Vorteile bietet DATEX-P nun gegenüber dem Telefon, welche Nachteile hat es?

Man kann zwar über Telefon das DATEX-P-Netz anwählen, jedoch ist die Umkehrung , d. h. aus DATEX-P heraus in das Telefonnetz zu gelangen, nicht möglich. Anwählbar sind einzig und allein Teilnehmer mit DATEX-P-Hauptanschluß bzw. über *Gateways*-Teilnehmer anderer Datenübertragungsnetze. Es kann derzeit keine der in der Bundesrepublik betriebenen Mb über DATEX-P erreicht werden, da diese nur über einen Telefonhauptanschluß verfügen. Wer sind dann aber die Teilnehmer von DATEX-P? Es sind vor allem Rechenzentren der Forschung, der Industrie und der öffentlichen Hand. Dieser Umstand macht DATEX-P zu einem Eldorado für Hacker, Cracker und Crasher. Die Verbindungsgebühren sind bei DATEX-P nicht entfernungsabhängig; sie richten sich im wesentlichen nach der übertragenen Datenmenge (Volumengebühr) und nach der Verbindungsdauer. Selbstverständlich gibt es auch hier wieder eine tageszeitabhängige Gebührenstaffel. Das „nicht entfernungsabhängig" gilt nur innerhalb des DATEX-P-Netzes. Bei der Einwahl in das Netz über Telefon sind die Gebühren für die Telefonverbindung vom Standort des Anrufers bis zum nächsten Übergangspunkt Telefon-DATEX-P (dem sog. Netzknoten) selbstverständlich entfernungsabhängig. Wer also das Pech hat, mehr als 100km vom nächsten Netzknoten entfernt zu wohnen, von denen 17 über die gesamte Bundesrepublik verteilt sind (s. **Bild 1**), für den lohnt sich zumindest innerhalb Deutschlands die Benutzung von DATEX-P finanziell nicht. Da die Benutzungsgebühren von DATEX-P sich aus sechs Einzelposten zusammensetzen, soll an dieser Stelle nicht näher darauf eingegangen werden.

Wie kann man nun DATEX-P Teilnehmer werden? Es besteht hier zum einen die Möglichkeit der Einrichtung eines DATEX-P-Hauptanschlusses, der allerdings sehr teuer und mit strengen Auflagen verbunden ist. Die zweite, kostengünstigere Möglichkeit ist die Beantragung einer Teilnehmerkennung (Network User Identification NUI). Mit dieser ist der Zugang über einen Netzknoten der DBP von einem Telefonhauptanschluß aus möglich. Die monatlichen Gebühren für die Teilnehmerkennung betragen DM 15.-. Antragsformulare sind vom Fernmeldeamt oder dem zuständigen Datennetzkoordinator erhältlich.

**Einige deutsche Mailboxen**
(mit Telefonnummer)

Decates 06154/51433
Dortmunder Mailbox 0231/170414
OTIS 06181/48884
Software Express 0211/414579
Saturn Mailbox 0221/1616284
Taunus Mailbox 06081/9677
Tedas 089/598423 und 089/596422
T.I.C. 030/7115078
TECOS 069/816787
WDR Computerclub 0221/371076

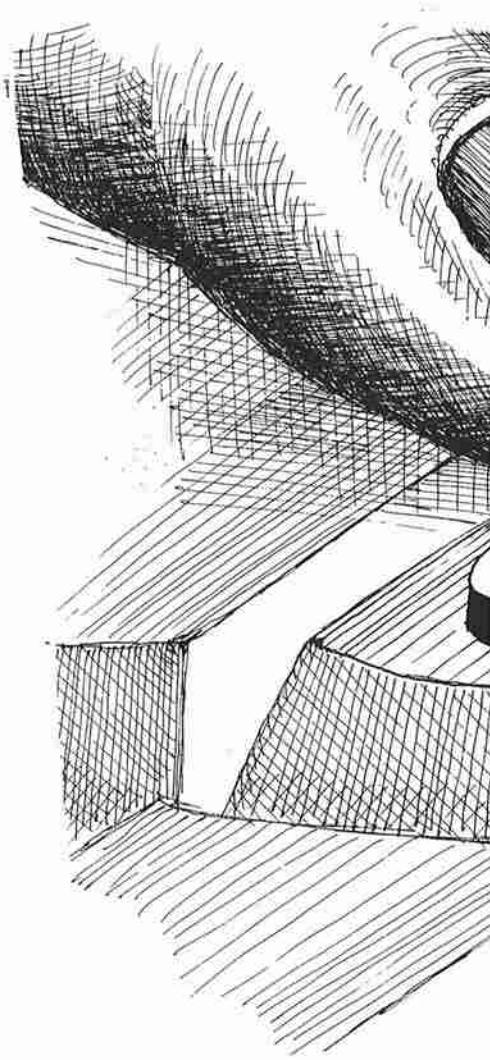
Dr. Jürgen B. Kehrel

# Ampersand macht's möglich

## INSTRING-Befehl in Applesoft

### 1. Grundlagen der Ampersand-Routinen

Obwohl Applesoft eine recht komfortable Sprache ist, besteht doch gelegentlich der Wunsch, einen Befehl, den Applesoft nicht kennt, auszuführen oder einfach etwas schneller zu erledigen. Als ich kürzlich in einem Programm nachprüfen mußte, ob einer oder mehrere von 77 kurzen Strings in einer 210 Zeichen langen Kette enthalten sind, dauerte das mit Stringvergleichen (MID$...) geschlagene 20 Minuten. Ein Maschinenprogramm erledigt nun die Aufgabe in 5 Sekunden.
Es gibt mehrere Möglichkeiten, Applesoft und Assembler in einem Programm zu benutzen, von denen wir uns zwei näher ansehen wollen: Die Ampersand-Routine und die CHRGET-Manipulation.



### Der Ampersand-Befehl (&)

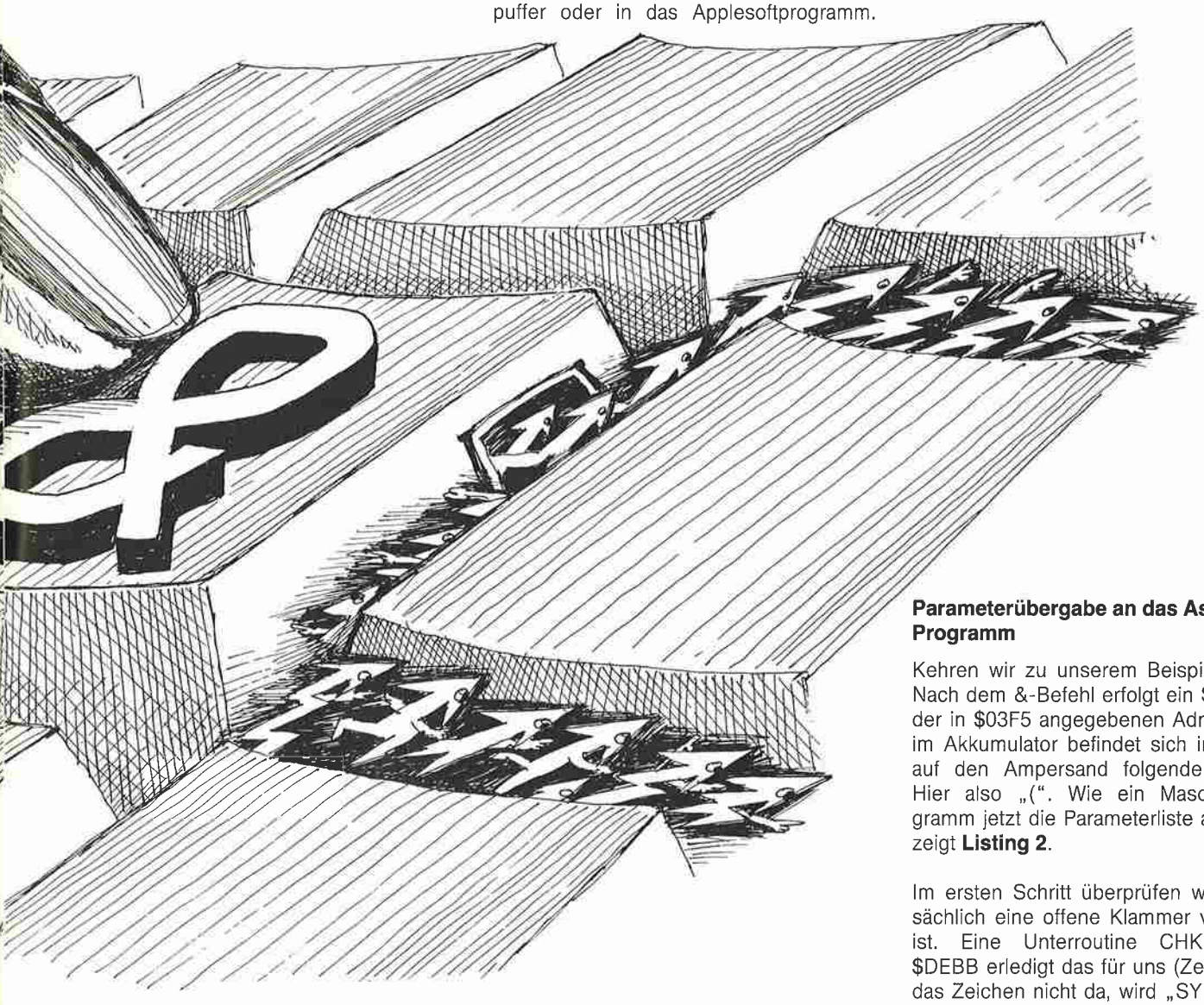
Immer wenn der Applesoftinterpreter das Zeichen & (= Ampersand) in einem Applesoft-Programm findet (nicht in Anführungszeichen eingeschlossen und nicht nach REM's), macht er einen Sprung zur Adresse $03F5. Dort steht normalerweise 4C 58 FF (JMP $FF58), ein Sprung zu einer festen RTS-Instruktion, die uns wieder ins laufende Applesoft-Programm zurückführt. Wir können nun ganz einfach, z.B. mit zwei POKE-Befehlen nach $03F6 und $03F7, den Sprung in unser Maschinenprogramm umlenken und von dort nach getaner Arbeit mit einem RTS in das Basic zurückkehren. In vielen Fällen ist uns damit schon gedient. Was aber, wenn wir Variablen zwischen beiden Programmteilen austauschen müssen? Der einfachste Weg ist, sie von Basic an feste Speicherplätze zu POKEn, vom Maschinenprogramm dort abzuholen, die Ergebnisse

dorthin zurückzuschreiben und mit PEEKs durch Applesoft abzufragen. Das ist aber nicht sehr elegant und erfordert für den Fall, daß ganze Strings übergeben werden müssen, viel Zeit und Speicherplatz. Der bessere Weg ist die Parameterübergabe. Kehren wir zu meinem obigen Problem zurück: Ich wollte wissen, ob B$ in A$ enthalten ist, wobei die Suche ab der Position I% in A$ beginnen sollte. Mein Ampersandbefehl erhält damit die Form & (A$, B$, I%).

**Die CHRGET-Routine**

Applesoft liest alle Befehle, gleichviel ob wir sie direkt eingeben oder sie sich in einem laufenden Programm befinden, mit Hilfe der CHRGET-Routine. Diese wird beim Einschalten des Apple aus dem ROM von $F10B nach $00B1 bis $00C8 kopiert. Schauen wir uns die disassemblierte Form einmal näher an (**Listing 1**).

Immer wenn Applesoft ein neues Zeichen lesen will, springt es nach CHRGET = $00B1 (Zeile 9). Zunächst werden $00B8 und gegebenenfalls auch $00B9, zusammen der sog. Textpointer, um 1 erhöht. Sie bilden im Low-High-Format die Argumente des LDA-Befehls in $00BA und weisen entweder in den Tastatureingabepuffer oder in das Applesoftprogramm. Das nächste Zeichen wird somit in den Akkumulator geladen und die restlichen Befehle (Zeile 13–20) nehmen noch einige Überprüfungen wahr. Will Applesoft dasselbe Zeichen noch einmal lesen, springt es direkt nach CHRGOT = $00B7, ohne den Textpointer zu verschieben. Von der Maschinensprache her können wir nun diese Routinen mitbenutzen sowie einige Teile des Applesoftinterpreters im ROM. Die einzige Schwierigkeit liegt nur darin, immer zu wissen, wohin der Textpointer gerade zeigt.



**Parameterübergabe an das Assembler-Programm**

Kehren wir zu unserem Beispiel zurück. Nach dem &-Befehl erfolgt ein Sprung zu der in $03F5 angegebenen Adresse, und im Akkumulator befindet sich immer das auf den Ampersand folgende Zeichen. Hier also „(". Wie ein Maschinenprogramm jetzt die Parameterliste abarbeitet, zeigt **Listing 2**.

Im ersten Schritt überprüfen wir, ob tatsächlich eine offene Klammer vorhanden ist. Eine Unterroutine CHKOPN bei $DEBB erledigt das für uns (Zeile 11). Ist das Zeichen nicht da, wird „SYNTAX ERROR" ausgegeben und das Programm gestoppt. Wird es gefunden, wird der Textpointer weitergesetzt und das nächste Zei-

chen in den Akkumulator geholt. Jetzt folgt unsere erste Variable A$, und wir müssen einen kleinen Exkurs machen, um zu verstehen, wie Variablen in Applesoft abgespeichert sind.
Einfache Variablen, z.B. I%, also keine Arrays wie B% (1), stehen im Speicher über dem Ende des Applesoftprogramms, also von LOMEM an aufwärts. Für jede Variable sind 7 Bytes vorhanden, die je nach Typ unterschiedlich belegt sind. Betrachten wir die für uns interessanten Fälle der Integer-Variablen (I%) und String-Variablen (A$).

Für I% = 264 finden wir vor:

1. 2. 3. 4. 5. 6. 7. Byte
C9 80 01 08 00 00 00

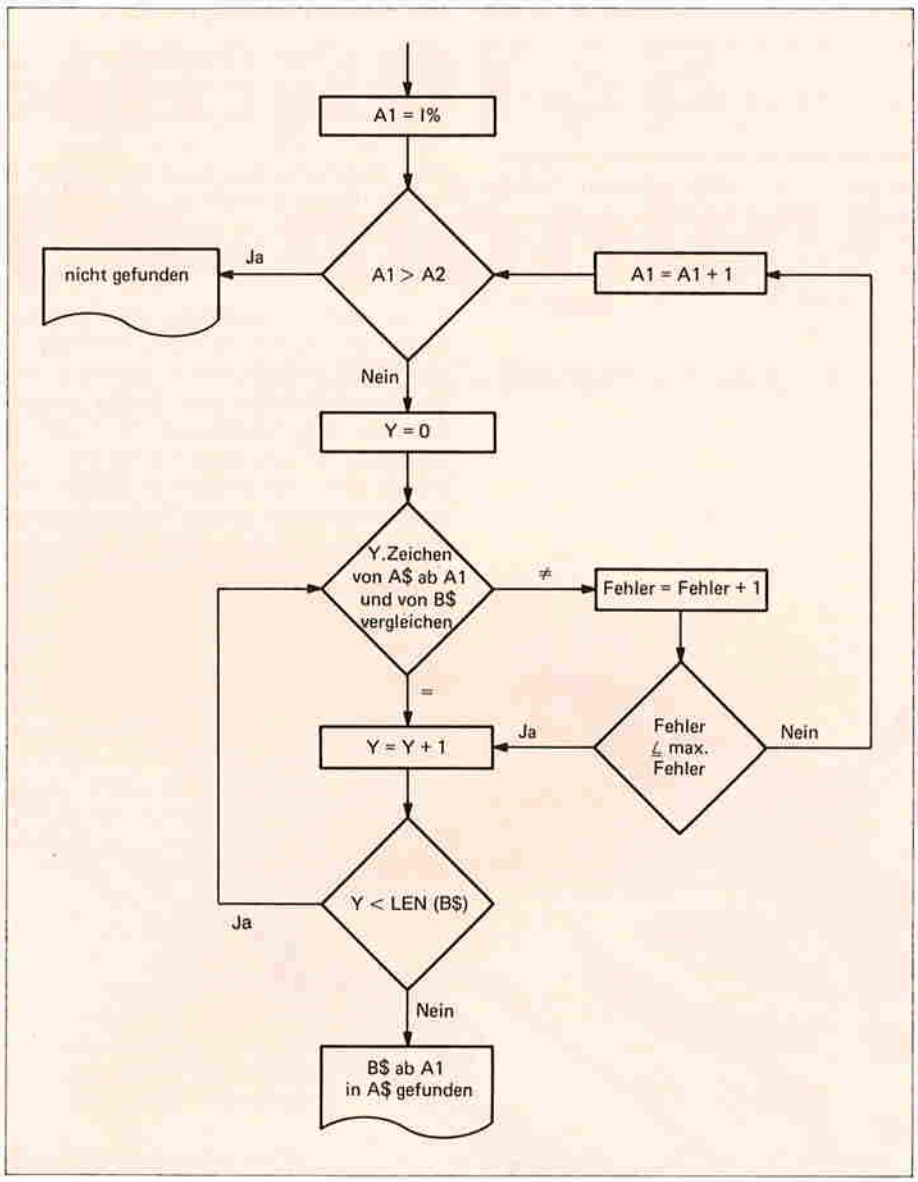
Byte 1 und 2 enthalten den Namen (maximal 2 Buchstaben) als ASCII-Werte mit gesetztem Bit 7 (> = $80). In Byte 3 und 4 steht unser Variablenwert, diesmal High-Low-Format ($0108 = 264). Die restlichen Bytes sind leer. Für A$ = „ABCDE" finden wir:

1. 2. 3. 4. 5. 6. 7. Byte
41 80 05 FB 95 00 00

Byte 1 und 2 enthalten den Variablennamen, allerdings mit Bit 7 im 1. Byte gelöscht, im 2. gesetzt. Ist der Variablenname nur ein Zeichen lang, wird $80 in das 2. Byte geschrieben. Das 3. Byte enthält die Stringlänge $05. Im 4. und 5. Byte finden wir jetzt nicht den String selber, sondern einen Zeiger zum ersten Zeichen des Strings (Low-High). Byte 6 und 7 sind leer. Applesoft speichert Strings unterhalb HIMEM, das in einem 48K Apple mit DOS 3.3 oder ProDOS normalerweise bei $9600 liegt. Die Strings stehen von dort abwärts, mit dem zuerst angelegten String ganz oben. Kurioserweise werden die einzelnen Zeichen eines Strings aber wieder aufwärts geschrieben. Wenn A$ unser erster String war, liest sich ein Speicherauszug so:

$95FB 41 = A
$95FC 42 = B
$95FD 43 = C
$95FE 44 = D
$95FF 45 = E

Byte 4 und 5 (unser Zeiger) weisen also auf das erste Zeichen unseres Strings.
Mit diesem Wissen im Hinterkopf kehren wir zu unserer Parameterübergabe zurück.




Um A$ auszuwerten, müssen wir nur noch wissen, wo genau die 7 Bytes abgespeichert sind. Glücklicherweise gibt es für diese Suche eine Routine namens PTRGET bei $DFE3 (Zeile 12), die wir aufrufen können. Sie findet A$ und schreibt die Adresse des 3. Bytes, also des Länge-Bytes, nach $0083 und $0084 (= VARPNT). Indirekt indiziert mit dem Y-Register lesen wir die Länge, speichern sie ab (Zeile 14, 15), und danach die Low-Adresse und die High-Adresse des Strings (Zeile 17–21).

Die PTRGET-Routine hat uns den Gefallen getan, den ganzen Stringnamen zu lesen, gleichviel wie lang er auch ist, so daß als nächstes das Komma folgt. Die Routine CHKCOM bei $DEBE wirkt wie CHKOPN.

Die zweite Variable wird genau wie A$ verarbeitet (Zeile 23–29), das Komma mit CHKCOM übersprungen. Es folgt I%, also eine Integer-Variable. Ihre 7-Byte-Information wird wie bei einem String gefunden, nur daß VARPNT diesmal direkt auf den Variablenwert zeigt. Mit Y = 1 indirekt indiziert erhalten wir das Low-Byte und speichern es ab (Zeile 34–37). Das High-Byte interessiert uns nicht, da im Zusammenhang mit Strings keine Länge größer als 255 vorkommt.

Wir haben jetzt alle Informationen, um unser Maschinenprogramm zu starten, das im nächsten Teil beschrieben ist. Wenden wir uns jetzt noch dem Problem zu, wie wir unser Ergebnis an das Applesoft-Programm übergeben.

### Parameterübergabe an das Applesoft-Programm

Unser Maschinenprogramm findet z.B. einen Teilstring B$ in A$ und speichert die Position in dem Speicherplatz POINT. Damit Applesoft diese Information erhält, übergeben wir den Wert an die Variable I%. Die Speicherplätze $0083 und $0084 (VARPNT) enthalten immer noch die Adresse von I%. Indirekt indiziert schreiben wir einfach den neuen Wert hinein (Zeile 40–42), so daß I% jetzt gleich POINT ist und wir ins Applesoft zurückkehren können. Ein einfaches RTS reicht aber nicht aus, da wir noch nicht die ganze Befehlszeile abgearbeitet haben, die geschlossene Klammer steht noch aus. Da uns diese jedoch nicht mehr interessiert, verschieben wir den Textpointer bis an den Anfang des nächsten Befehls und springen nach Applesoft. Die Routine DATA bei $D995 besorgt das alles für uns.

## 2. INSTRING-Suche

Nachdem wir im ersten Teil gesehen haben, wie wir mit dem Ampersand-Befehl & Maschinenprogramme aufrufen und anschließend Parameter übergeben können, nutzen wir unsere neuen Möglichkeiten jetzt an einem ausgearbeiteten Beispiel. Das Problem ist noch dasselbe: Finden eines kurzen Strings B$ in einem langen String A$. Wir erweitern die Aufgabe noch etwas, indem wir nicht nur sagen, ab welchem Zeichen I% in A$ die Suche beginnen soll, sondern zusätzlich vorgeben, wieviele Fehler dabei zugelassen sind, um nicht nur genau passende (F% = 0) sondern auch ähnliche Teile von A$ (F% > 0) zu finden. F% sollte natürlich nie so groß sein wie B$ lang ist, sonst wird einfach alles gefunden. Die Routine soll nach dem Ablauf die Position der Fundstelle in I% und die tatsächliche Fehlerzahl, die ja kleiner als die erlaubte sein kann, in F% enthalten. Wird nichts gefunden, werden beide Variablen auf 0 gesetzt. **Listing 3** zeigt das vollständige Programm als LISA 2.5 Assembler-Source. Es wurden keine speziellen Opcodes benutzt außer EPZ (= Equals Page Zero), was bei anderen Assemblern wie z.B. Big Mac in EQU umzuwandeln ist. Wer keine Assembler benutzt, kann auch den HEX-Code direkt eingeben. Gehen Sie dazu mit CALL -151 in den Monitor. Hinter das erscheinende Sternchen schreiben Sie die Anfangsadresse 9400, gefolgt von einem Doppelpunkt. Direkt dahinter schreiben Sie die erste HEX-Zahl, dann drücken Sie die

Listing 1

```
00B8              7  TXTPTR   EPZ $B8
00B1              8  *
00B1 E6 B8        9  CHRGET   INC TXTPTR
00B3 D0 02       10           BNE CHRGOT
00B5 E6 B9       11           INC TXTPTR+1
00B7 AD FF FF    12  CHRGOT   LDA $FFFF          ;$FFFF NICHT WIRKLICH
00BA C9 3A       13           CMP #$3A           ;IST ES ":",DANN
00BC B0 0A       14           BCS RET            ;Z UND C SETZEN
00BE C9 20       15           CMP #$20           ; IST ES " ", DANN
00C0 F0 EF       16           BEQ CHRGET         ;NAECHSTES ZEICHEN
00C2 38          17           SEC
00C3 E9 30       18           SBC #$30           ;C LOESCHEN FALLS ZAHL
00C5 38          19           SEC
00C6 E9 D0       20           SBC #$D0           ;C SETZTEN FALLS N. ZAHL
00C8 60          21  RET      RTS                ;ZURUECK
```

Listing 2

```
0800             10  *
0800 20 BB DE    11           JSR CHKOPN         ;IST ES "("?
0803 20 E3 DF    12           JSR PTRGET         ;HOLE ADRESSE DES OBJ.STRINGS
0806 A0 00       13           LDY #0
0808 B1 83       14           LDA (VARPNT),Y     ;HOLE LAENGEN-BYTE
080A 8D 45 08    15           STA LOBJ           ;SPEICHERE ES
080D C8          16           INY
080E B1 83       17           LDA (VARPNT),Y     ;HOLE LO-BYTE DES ZEIGERS
0810 85 3C       18           STA A1             ;UND SPEICHERE ES
0812 C8          19           INY
0813 B1 83       20           LDA (VARPNT),Y     ;HOLE HI-BYTE
0815 85 3D       21           STA A1+1           ;SPEICHERE ES
0817 20 BE DE    22           JSR CHKCOM         ;IST ES ","?
081A 20 E3 DF    23           JSR PTRGET         ;HOLE ADRESSE DES TARGET-STRINGS
081D A0 00       24           LDY #0
081F B1 83       25           LDA (VARPNT),Y     ;HOLE LAENGEN-BYTE
0821 8D 46 08    26           STA LTAR           ;SPEICHERE ES
0824 C8          27           INY
0825 B1 83       28           LDA (VARPNT),Y     ;HOLE LO-BYTE DES ZEIGERS
0827 85 40       29           STA A3             ;UND SPEICHERE ES
0829 C8          30           INY
082A B1 83       31           LDA (VARPNT),Y     ;HOLE HI-BYTE DES ZEIGERS
082C 85 41       32           STA A3+1           ;SPEICHERE ES
082E 20 BE DE    33           JSR CHKCOM         ;IST ES ","?
0831 20 E3 DF    34           JSR PTRGET         ;HOLE ADRESSE DER %VARIABLEN
0834 A0 01       35           LDY #1
0836 B1 83       36           LDA (VARPNT),Y     ;HOLE LO BYTE
0838 8D 47 08    37           STA POINT
083B             38  * HIER FOLGT DAS HAUPTPROGRAMM *
083B             39  ********************************
083B A0 01       40           LDY #1
083D AD 47 08    41           LDA POINT
0840 91 83       42           STA (VARPNT),Y     ;SPEICHERE POSITION IN %VARIABLE
0842 4C 95 D9    43           JMP DATA           ;SPRINGE ZURUECK NACH APPLESOFT
```

Listing 3

```
0800              3           TTL "INSTRING"
0800              4  *
0800              5  * SUCHT EINEN SUBSTRING (TARGET) IN
0800              6  * EINEM OBJEKTSTRING, WOBEI DIE UEBER-
0800              7  * EINSTIMMUNG EINE VORHER ANGEGEBENE
0800              8  * ZAHL VON FEHLERN (UNVOLLKOMMENE UEBER-
0800              9  * EINSTIMMUNG) ENTHALTEN DARF. GIBT ALS
0800             10  * ERGEBNIS DEN BEGINN DER UEBEREINSTIM-
0800             11  * MUNG (0 = NICHT GEFUNDEN) UND DIE
0800             12  * ZAHL DER FEHLER AUS. AUFRUF DER ROUTINE
0800             13  * MIT &(OBJ$,TAG$,FE%,ST%)
0800             14  * WOBEI FE% UND ST% BEI DER EINGABE
0800             15  * MAX. FEHLERZAHL UND START DER SUCHE
0800             16  * ANGEBEN UND NACH DEM PROGRAMMABLAUF
0800             17  * DIE ERREICHTE FEHLERZAHL UND DEN BE-
0800             18  * GINN DER UEBEREINSTIMMUNG ENTHALTEN
0800             19  *
0800             20  * VON JUERGEN B. KEHREL AUGUST 1984
0800             21  *
0800             22  ****************************************
0800             23  *
0006             24  VAR      EPZ $6
003C             25  A1       EPZ $3C
003E             26  A2       EPZ $3E
0040             27  A3       EPZ $40
0073             28  HIMEM    EPZ $73
0083             29  VARPNT   EPZ $83
```

Leertaste, geben die zweite HEX-Zahl ein, wieder die Leertaste usw., so daß Sie folgendes Bild erhalten:

*9400:A9 4C 8D F5 03 ...

Nach spätestens 85 HEX-Zahlen drücken Sie die Return-Taste, am besten schon, wenn Sie zwei bis drei Bildschirmzeilen eingegeben haben. Es erscheint wieder der Stern. Nun geben Sie gleich den Doppelpunkt (der Monitor weiß selbst die richtige Adresse) und weitere HEX-Zahlen ein, bis alles eingetippt und mit Return abgeschlossen ist. Drücken Sie dann die Control-Taste und C (= CTRL-C) gleichzeitig, gefolgt von Return, und Sie sind zurück im Basic. Speichern Sie das Programm mit dem Befehl BSAVE INSTRING, A$9400, L$00FA auf Diskette ab.
Das Programm besteht aus zwei Teilen: Zeile 44–56 und 57–158. Der erste Teil wird nur einmal zur Initialisierung des Systems benutzt. Sie sollten von Ihrem Applesoft-Programm aus (**Listing 4 oder 5**), noch bevor Sie irgendeine Stringvariable anlegen, das Maschinenprogramm starten, z.B. mit
10 PRINT CHR$(4) „BRUN INSTRING".
Dadurch wird ein Sprung nach $942C in den Ampersand-Vektor bei $03F5 geschrieben sowie HIMEM unter das Maschinenprogramm gelegt, so daß es nicht überschrieben werden kann. (Dieses Verfahren gilt nur für DOS 3.3, nicht für ProDOS, da bei ProDOS im Falle einer HIMEM-Änderung zusätzlich die System Bit Map geändert werden muß. Anm. der Red.)
Wird das Programm jetzt von Applesoft über Ampersand aufgerufen, erfolgt der Einstieg immer an der zweiten Einsprungstelle bei $942C. Hier erkennen wir die schon im ersten Teil besprochene Parameterübernahme wieder, die leicht modifiziert wurde. In Zeile 82–86 werden die Längen von A$ und B$ verglichen und eine Fehlermeldung „SYNTAX ERROR" ausgegeben, falls B$ > A$ ist. Sodann wird die Längendifferenz zum Beginn von A$ hinzuaddiert und nach A2 gespeichert, um das Such-Ende festzulegen. Da diesmal zwei Integer-Variablen gelesen und auch wieder zurückgespeichert werden müssen, wird der Zeiger zu F% nach VAR zwischengespeichert; der zu I% bleibt in VARPNT erhalten. In Zeile 144–148 wird in umgekehrter Reihenfolge, erst I%, dann F%, das Ergebnis zurückgespeichert. Wegen der indirekt indizierten Adressierung mußte VAR eine Zero Page Adresse sein.

```
DFE3              30  PTRGET   EQU $DFE3
DEBE              31  CHKCOM   EQU $DEBE
DEBB              32  CHKOPN   EQU $DEBB
DEC9              33  SYNERR   EQU $DEC9
D995              34  DATA     EQU $D995
FCBA              35  NXTA1    EQU $FCBA
03F5              36  AMPER    EQU $3F5
0800              37  *
9400              38           ORG $9400
9400              39           OBJ $1400
9400              40  *
9400              41  * ERSTER EINSTIEG HIER
9400              42  * SETZT HIMEM: UND AMPERJUMP &
9400              43  *
9400 A9 4C        44  START    LDA #$4C
9402 8D F5 03     45           STA AMPER
9405 A9 2C        46           LDA #CALL
9407 8D F6 03     47           STA AMPER+1
940A A9 94        48           LDA /CALL
940C 8D F7 03     49           STA AMPER+2
940F A9 00        50           LDA #START
9411 85 73        51           STA HIMEM
9413 A9 94        52           LDA /START
9415 85 74        53           STA HIMEM+1
9417 D8           54           CLD
9418 60           55           RTS
9419 D6 CF CE     56           ASC "VON 1984 J.B.KEHREL"
941C A0 B1 B9
941F B8 B4 A0
9422 CA AE C2
9425 AE CB C5
9428 C8 D2 C5
942B CC
942C              57  *
942C              58  * APPLESOFT SPRINGT HIERHIN
942C              59  *
942C 20 BB DE     60  CALL     JSR CHKOPN          ;IST ES "("?
942F 20 E3 DF     61           JSR PTRGET          ;HOLE ADRESSE DES OBJ.STRINGS
9432 A0 00        62           LDY #0
9434 B1 83        63           LDA (VARPNT),Y      ;HOLE LAENGEN-BYTE
9436 8D F5 94     64           STA LOBJ            ;SPEICHERE ES
9439 C8           65           INY
943A B1 83        66           LDA (VARPNT),Y      ;HOLE LO-BYTE DES ZEIGERS
943C 85 3C        67           STA A1              ;UND SPEICHERE ES
943E C8           68           INY
943F B1 83        69           LDA (VARPNT),Y      ;HOLE HI-BYTE
9441 85 3D        70           STA A1+1            ;SPEICHERE ES
9443 20 BE DE     71           JSR CHKCOM          ;IST ES ","?
9446 20 E3 DF     72           JSR PTRGET          ;HOLE ADRESSE DES TARGET-STRINGS
9449 A0 00        73           LDY #0
944B B1 83        74           LDA (VARPNT),Y      ;HOLE LAENGEN-BYTE
944D 8D F6 94     75           STA LTAR            ;SPEICHERE ES
9450 C8           76           INY
9451 B1 83        77           LDA (VARPNT),Y      ;HOLE LO-BYTE DES ZEIGERS
9453 85 40        78           STA A3              ;UND SPEICHERE ES
9455 C8           79           INY
9456 B1 83        80           LDA (VARPNT),Y      ;HOLE HI-BYTE DES ZEIGERS
9458 85 41        81           STA A3+1            ;SPEICHERE ES
945A 38           82           SEC
945B AD F5 94     83           LDA LOBJ            ;IST OBJ. > TARGET ?
945E ED F6 94     84           SBC LTAR
9461 B0 03        85           BCS OK
9463 4C C9 DE     86           JMP SYNERR          ;TARGET > OBJECT
9466 18           87  OK       CLC                 ;SETZE A2 AUF DIE LETZTE
9467 65 3C        88           ADC A1              ;VERGLEICHSPOSITION
9469 85 3E        89           STA A2
946B A5 3D        90           LDA A1+1
946D 69 00        91           ADC #0
946F 85 3F        92           STA A2+1
9471 20 BE DE     93           JSR CHKCOM          ;IST ES ","?
9474 20 E3 DF     94           JSR PTRGET          ;HOLE ADRESSE DER 1. VARIABLEN
9477 A5 83        95           LDA VARPNT          ;HOLE LO BYTE ADRESSE 1.VAR
9479 85 06        96           STA VAR             ;UND SPEICHERE SIE
947B A5 84        97           LDA VARPNT+1        ;HOLE HI BYTE ADRESSE 1.VAR
947D 85 07        98           STA VAR+1           ;UND SPEICHERE SIE
947F A0 01        99           LDY #1
9481 B1 83       100           LDA (VARPNT),Y      ;HOLE LO BYTE WERT
9483 8D F9 94    101           STA MAXM            ;SPEICHERE MAX. FEHLERZAHL
9486 20 BE DE    102           JSR CHKCOM          ;IST ES ","?
9489 20 E3 DF    103           JSR PTRGET          ;HOLE ADRESSE DER 2. VARIABLEN
948C A0 01       104           LDY #1
948E B1 83       105           LDA (VARPNT),Y      ;HOLE LO BYTE
9490 D0 03       106           BNE POS
9492 4C C9 DE    107           JMP SYNERR          ;START BEI 0 NICHT MOEGLICH
9495 8D F7 94    108  POS      STA POINT
9498 CE F7 94    109           DEC POINT
949B 18          110           CLC
949C A5 3C       111           LDA A1
```

## "PAPPLEN" Sie Ihren Drucker auf!

Kennen Sie das nicht auch?!
Sie brauchen ein Zeichen und es ist nicht da.

☎ ⇒ ξ ♡ ξ Σ ∫xₙdx γ ↰
ᵏ ⊠ ✉ μ ☁ ⚸ ⚸

Der DMP Charger liefert die Lösung.

Mühelos können Sie Zeichen selbst erstellen und zum Apple Imagewriter oder DMP übertragen. Die Zeichen sind dann im Drucker gespeichert. In Ihrem Textprogramm stehen diese Zeichen mit Hilfe einer ESCAPE-Sequenz zur Verfügung. Das Programm arbeitet auf allen Apple//Typen und Kompatiblen (64 K).

Preis: DM 198,– incl. MwSt. und ausf. Handbuch
Übungsdiskette für ein Zeichen: DM 12,50
Versand gegen Vorauskasse oder NN durch:

**Norbert Hunstig**
✉ Nottulner Landweg 81
D-4400 Münster
☏ 0 25 34 / 74 49



## cvb computer

Repsoldstraße 49, 2000 Hamburg 1, Tel 040-230685

### Komplettsystem SUPER II

+ 48 KB Mikrocomputer, Kleinschreibung, deutsche Tastatur, Zehnerblock, Auto-Repeat, APPLE kompatibel, 8 Slots
+ 16 KB Erweiterung auf 64 KB Hauptspeicher (Sprachkarte), PASCAL-, VISICALC- und CP/M-kompatibel, busgepuffert
+ 5.25"-Diskettenlaufwerk TEAC FD 55-A, voll kompatibel zu APPLE-Laufwerken, 143 KB, mit Controller-Karte
+ 12"-Datenmonitor SANYO DM 2112 CX, grüner Bildschirm, 15 MHz Auflösung, geätzte Bildröhre

anschlußfertig DM 2450.00

+ TEAC 5.25"-Laufwerk FD 55-A
+ Z 80-Mikroprozessorkarte
+ 80 Zeichen-Superkarte VIDEX kompatibel, deutscher und ASCII-Zeichensatz, Grafik, Softswitches, busgepuffert

DM 1100.00

### TEAC 55-F

Doppellaufwerk 2x 640 KB mit Gehäuse, Controller, Kabel und Software, anschlußfertig für Äpfel

DM 2198.00

### SPEEDY 100

80 cps Matrixdrucker A4, Einzelblatt und Traktor, Grafik, EPSON-kompatibel Centronics Parallel

DM 799.00

### JUKI 6100

Profi-Typenraddrucker A4 quer, 22 cps, 2K Puffer, WordStar kompatibel, Centronics Parallel

DM 1699.00

Apple · NCR · Epson · JUKI · Speedy · C.Itoh · NEC · Olympia · Sanyo
Taxan · Teac · Verbatim · Scotch 3M · M+T · Zubehör · Bücher
Barpreise incl. Mehrwertsteuer · volle Garantie
Abholung oder Versand · Infoheft anfordern



# FORTH - SYSTEME

| | | |
|---|---|---|
| FORTH Encyclopedia | DM | 98,–– |
| System Guide to FIG FORTH | DM | 98,–– |
| Floating Point Listing in FORTH | DM | 70,60 |
| Expertensystem in FORTH 79 | DM | 98,–– |
| Dokumentation der FORTH Konferenzen 1980, 240 Seiten, 1982, 300 Seiten, | je DM | 98,–– |
| 1981, 600 Seiten (2 Bände) | DM | 156,80 |
| Rochester FORTH Proceedings 1981, 1982, 1983 | je DM | 98,–– |
| Z80 FORTH von LMI auf APPLE-Disk | DM | 376,20 |
| APPLE 3D-GraFORTH | DM | 282,20 |
| APPLE II Master FORTH mit Editor, Assembler, Debugger, 250 S. Handbuch | DM | 398,–– |
| MasterFORTH + Hires + Floatingp. | DM | 598,–– |
| Expertensystem auf Diskette | DM | 376,20 |
| fysFORTH mit Einführungskurs | DM | 98,–– |
| Professional Applikation Development System mit Metacompiler, FPA | DM | 1881,–– |
| Cross Compiler für LMI Systeme | DM | 1250,–– |

Wir unterstützen natürlich auch andere Rechner, fordern Sie doch einfach unseren kostenlosen Katalog an:

**FORTH-Systeme Angelika Flesch, Schützenstr. 3, 7820 Titisee-Neustadt, Telefon 0 76 51 / 16 65**



APPLE kompatible

Erweiterungen, die noch niemand anzubieten hat

AKUSTIKKOPPLER — auf APPLE-Slot steckbar 249,-
mit Communication-Software
optionale V24-Schnittstelle

GRAFIK — RGB, 8 Farben, 8-fach Zoom
verschiedene Zeichensätze
eigener Bildschirm - Speicher
auf APPLE-Slot steckbar

FARB-MONITOR — 14" mittel- und hochauflösend
Horizontalfrequenz 32 kHz
10,5 MHz und 31,6 MHz

**edicta**
Lindenstrasse 25
6290 Weilburg 4
Phon 06471/2473

Die Suche verläuft nach dem abgebildeten **Flußdiagramm** INSTRING ab. Sie ist in Zeile 124–135 codiert, wobei wir uns die Monitor-Routine NXTA1 bei $FCBA zunutze machen, um A1 hochzusetzen. Wenn A1 vor der Addition von 1 schon > = A2 war, wird das Carry-Bit im Prozessor gesetzt, was mit dem folgenden BCS (Branch on Carry Set) überprüft wird, und das erfolglose Ende der Suche angezeigt. POINT und MISM werden auf 0 gesetzt (Zeile 141–143) und ausgegeben. Zeile 136–140 ist die Überprüfung, ob schon mehr Fehler vorliegen, als erlaubt sind. Wenn ja, wird die Suche an dieser Stelle abgebrochen und eine Position in A$ weitergerückt. Wenn nein, geht der Vergleich weiter. Wird bis zur vollen Länge von B$ Übereinstimmung erzielt (Zeile 132–133), werden Position und Fehlerzahl ausgegeben (Zeile 144–149). Unsere Routine ist damit komplett. Sie sucht natürlich nur bis zur ersten Fundstelle, aber das umgebende Basicprogramm kann ja die Suche wieder aufnehmen lassen, falls A$ noch genügend lang ist. Das folgende Applesoft-Demo zeigt, wie das bewerkstelligt wird (**Listing 4**). Dabei ist darauf zu achten, daß die erlaubte Fehlerzahl immer wieder nach F% zurückgeschrieben wird (Zeile 70), da ja das Maschinenprogramm auch F% verändert.

Falls Sie in einem großen Programm nach vielen Strings suchen wollen, schreiben Sie eine FOR-NEXT Schleife (**Listing 5**) und legen die Ergebnisse in vorher dimensionierte Variablen ab.

### Zusammenstellung der Interpreter-Routinen

CHRGET = $00B1 = erhöht Textpointer um 1 und lädt Akkumulator mit Textpointer-Byte.

CHRGOT = $00B7 = lädt Akkumulator mit Byte von Textpointer-Stelle, ohne Textpointer zu erhöhen.

DATA = $D995 = rückt Textpointer zum nächsten Doppelpunkt oder zum Ende der Zeile (End of Line) vor.

PTRGET = $DFE3 = liest Variablenname von Textpointer-Stelle und sucht dann Speicheradresse des Wertes der Variablen. Danach ist das Low-Byte der Speicheradresse im Akkumulator und das High-Byte im Y-Register. Ferner wird der Zero-Page-Pointer $0083 beeinflußt (s. Aufsatz).

CHKOPN = $DEBB = prüft, ob Textpointer-Byte „(“ ist.

CHKCOM = $DEBE = prüft, ob Textpointer-Byte „,“ ist.

```
949E 6D F7 94   112          ADC POINT          ;SETZE A1 AUF DEN WERT
94A1 85 3C      113          STA A1             ;VOR VERGLEICHSBEGINN
94A3 A5 3D      114          LDA A1+1
94A5 69 00      115          ADC #0
94A7 85 3D      116          STA A1+1
94A9 38         117          SEC
94AA A5 3C      118          LDA A1
94AC E9 01      119          SBC #1
94AE 85 3C      120          STA A1
94B0 A5 3D      121          LDA A1+1
94B2 E9 00      122          SBC #0
94B4 85 3D      123          STA A1+1
94B6 20 BA FC   124  NEXT    JSR NXTA1          ;ERHOEHE A1
94B9 B0 23      125          BCS NOTFND         ;SUBSTRING NICHT GEFUNDEN
94BB EE F7 94   126          INC POINT
94BE A0 00      127          LDY #0
94C0 8C F8 94   128          STY MISM
94C3 B1 3C      129  LOOP1   LDA (A1),Y         ;HOLE OBJ BYTE
94C5 D1 40      130          CMP (A3),Y         ;VERGLEICHE MIT TARGET BYTE
94C7 D0 08      131          BNE ERR            ;NICHT IDENTISCH
94C9 C8         132  LOOP2   INY                ;GEHE ZUM NAECHSTEN BYTE
94CA CC F6 94   133          CPY LTAR           ;IST ES LETZTES TARGET BYTE?
94CD D0 F4      134          BNE LOOP1          ;NEIN, WEITER!
94CF F0 15      135          BEQ HIT            ;GEFUNDEN!
94D1 EE F8 94   136  ERR     INC MISM
94D4 AD F9 94   137          LDA MAXM
94D7 CD F8 94   138          CMP MISM
94DA B0 ED      139          BCS LOOP2          ; MAXM>=MISM, WEITER
94DC 90 D8      140          BCC NEXT           ;MAXM < MISM, NICHT GEFUNDEN
94DE A9 00      141  NOTFND  LDA #0
94E0 8D F7 94   142          STA POINT          ;LOESCHE DIE PARAMETER
94E3 8D F8 94   143          STA MISM
94E6 A0 01      144  HIT     LDY #1
94E8 AD F7 94   145          LDA POINT
94EB 91 83      146          STA (VARPNT),Y     ;SPEICHERE POSITION IN 2. VARIABLE
94ED AD F8 94   147          LDA MISM
94F0 91 06      148          STA (VAR),Y        ;SPEICHERE NR. DER MISMATCHES
94F2 4C 95 D9   149          JMP DATA           ;SPRINGE ZURUECK NACH APPLESOFT
94F5            150  *
94F5            151  * DATEN SPEICHER
94F5            152  *
94F5            153  LOBJ    DFS 1,0
94F6            154  LTAR    DFS 1,0
94F7            155  POINT   DFS 1,0
94F8            156  MISM    DFS 1,0
94F9            157  MAXM    DFS 1,0
00FA            158  ZEND    EQU *-START
```

Listing 4

```
 10  PRINT  CHR$ (4);"BRUN INSTRING"
 20  HOME : CLEAR
 30  INPUT "GEBEN SIE DEN HAUPTSTRING EIN ";A$
 40  PRINT : INPUT "GEBEN SIE DEN SUCHSTRING EIN ";B$
 50  PRINT : INPUT "WIEVIELE FEHLER MAXIMAL? ";FE%
 60  PRINT : INPUT "AB WELCHER POSITION SUCHEN? ";I%
 70 F% = FE%
 80  & (A$,B$,F%,I%)
 90  PRINT : PRINT "DER SUCHSTRING WURDE AB DER ";I%;"."
100  PRINT "POSITION GEFUNDEN MIT ";F%;" FEHLERN"
110  IF I% <  LEN (A$) -  LEN (B$) THEN I% = I% + 1:
     GET C$: GOTO 70
120  END
```

Listing 5

```
 10 PRINT  CHR$ (4);"BRUN INSTRING": REM
    Hier steht Ihr Programm, in dem
    alle nötigen Strings dimensio-
    niert werden und ein Array C$(1)
    C$(2)...C$(n) gebildet wird.
100 NT = 0: REM Zähler
110  FOR I = 1 TO N
120 B$ = C$(I)
130 I% = 1
140 F% = FE%
150  & (A$,B$,F%,I%)
160  IF I% = 0 THEN 220
170 NT = NT + 1
180 PO%(NT) = I%: REM Position
190 FL%(NT) = F%: REM Fehler
200 ST%(NT) = I: REM String Nr.
210  IF I% <  LEN (A$) -  LEN (B$) THEN I% = I% + 1: GOTO 140
220  NEXT I
230  REM Hier geht Ihr Programm weiter.
```

# NEU soeben erschienen

## Apple ProDOS für Aufsteiger, Band 1

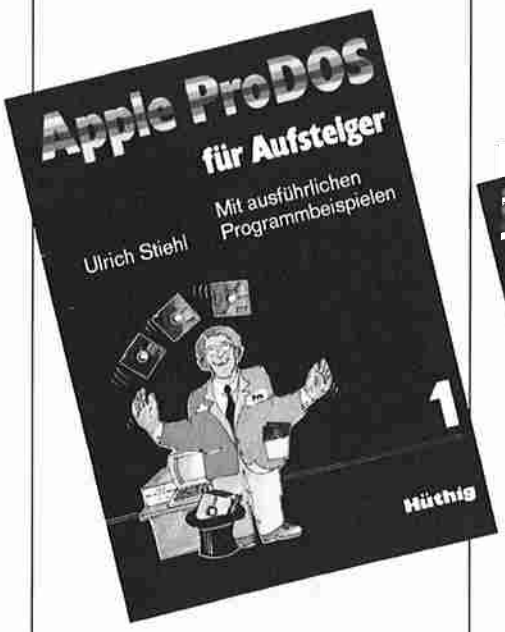
**Mit ausführlichen Programmbeispielen**

**von U. Stiehl**

1984, 208 S., kart., DM 28,–

Begleitdiskette zu „Apple ProDOS für Aufsteiger", DM 28,–
ISBN 3-7785-1032-0

ProDOS ist das neue Betriebssystem für den Apple II Plus/IIe. Applesoft-Programmierer, die unter DOS 3.3 gearbeitet haben, werden sich schnell an ProDOS gewöhnen, da ProDOS und DOS 3.3 in dieser Hinsicht weitgehend kompatibel sind. Dagegen müssen Assembler-Programmierer völlig umdenken. Deshalb liegt das Schwergewicht von „Apple ProDOS für Aufsteiger, Band 1" auf der Assemblerprogrammierung und der minutiösen Darstellung der ProDOS-internen Systemadressen, die jedoch auch für Applesoft-Programmierer von großer Bedeutung sind.



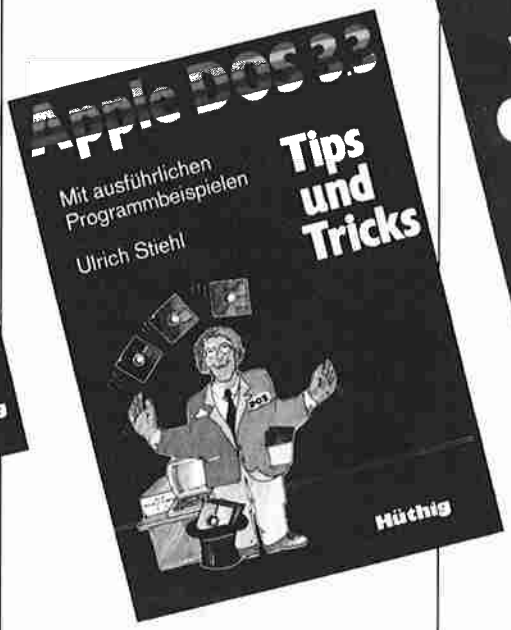

## Apple DOS 3.3 – Tips und Tricks

**von U. Stiehl**

1984, 216 S., mit zahlreichen, ausführlich kommentierten Programmlistings, kart., DM 28,–

Begleitdiskette zu „Apple DOS 3.3", DM 28,–

Dies ist die erste deutschsprachige Darstellung des Diskettenbetriebssystems DOS 3.3 für den Apple II/II Plus/IIe, die sich sowohl an Applesoft- als auch an Assembler-Programmierer wendet.
Das Buch behandelt ausführlich die dem Applesoft-Programmierer zur Verfügung stehenden DOS-Befehle, wobei die Textfiles wegen ihrer großen Bedeutung besonders eingehend dargestellt werden. Für den Assembler-Programmierer enthält es neben einer detaillierten Beschreibung der DOS-Interna elf vollständige RWTS-Anwendungsprogramme – z. B. CPM-Refiner, DOS-lose Datendisk, TSL-Maker, File-Reader, Pseudo-Disk-Driver und Fastbrun-Routine –, die Techniken enthüllen, die bislang noch niemals publiziert waren.




## dBASE II

**von Wolfgang Eggerichs**

**Band 1: Einführung**

1984, 174 S., kart., DM 39,80

Diese Buchreihe befaßt sich mit dem Datenbanksystem dBASE II, einem speziell für Mikrocomputer entwickelten System. Dieses Datenbanksystem läuft unter den Betriebssystemen CP/M, MP/M, MS-DOS und PC-DOS.
Um dem Anfänger den Einstieg in dieses doch recht mächtige Software-Werkzeug zu erleichtern, werden in den beiden ersten aufeinander abgestimmten Bänden jeweils die zu einem bestimmten Leistungsbereich gehörenden Kommandos herausgefiltert und erläutert. Zusätzlich sind kleine Aufgaben integriert, an denen der Leser seinen Kenntnisstand von dBASE II überprüfen kann.

VERLAGSGRUPPE
**Hüthig**

06221/489353

6900 Heidelberg · Postfach 102869

# IF-THEN-ELSE-Befehl simuliert in Applesoft

**Franz-Josef Hüskens**

In der Programmiersprache Applesoft-Basic ist die Anweisung IF-THEN-ELSE nicht implementiert. Im folgenden Beitrag wird eine Möglichkeit dargestellt, mit den vorhandenen Lauf- bzw. Kontrollanweisungen diesen Befehl zu simulieren.

Ein Basic-Programm besteht aus einer Reihe von Anweisungen, die dem Computer die Aufgabe darstellen, die er zu lösen hat. Jede Anweisung, die innerhalb des Programms durchgeführt werden soll, muß in einer Programmzeile stehen. Im Applesoft-Basic ist es jedoch auch möglich, mehrere Befehle in eine Zeile zu schreiben. Allerdings muß dann jeder Einzelbefehl durch einen Doppelpunkt von der vorhergehenden Anweisung und, falls erforderlich, mit einem weiteren Doppelpunkt von der nachfolgenden Anweisung getrennt werden. Dieses „Zusammenpakken" mehrerer Programmzeilen bietet den Vorteil der Speicherplatzersparnis; außerdem werden die Programme in der Ausführung schneller. Dabei ist logisch, daß in einem umfangreichen Programm mehr Speicherplatz gespart werden kann und der Zeitgewinn erheblich höher ist.
Die Programmzeilennummern regeln die Abfolge der durchzuführenden Befehle. Ist ein Programm ohne Anweisungen geschrieben, die diesen Ablauf ändern, so werden zuerst die Befehle der ersten Zeile durchgeführt, dann die der zweiten usw.
Derartige sequentielle Programme (siehe **Beispiel 1**) sind jedoch sehr selten. In fast allen Programmen muß der durch die Programmzeilennummern festgelegte Ablauf des öfteren geändert werden. Das kann mit den im folgenden kurz beschriebenen Anweisungen geschehen:

### 1. GOTO-Anweisung

Der GOTO-Anweisung muß als Argument eine Zeilennummer folgen, zu der dann der weitere Programmablauf verzweigt. Als nächste Anweisung wird der erste Befehl der angesprungenen Programmzeile durchgeführt. Im **Beispiel 2** verzweigt das Programm von Zeile 35 zur Zeile 50; die Anweisung in Programmzeile 40 wird nicht ausgeführt.

### 2. GOSUB-Anweisung

Der GOSUB-Anweisung folgt als Argument ebenfalls eine Programmzeilennummer, zu der dann das Hauptprogramm verzweigt. Der Unterschied zur GOTO-Anweisung besteht jedoch darin, daß mit diesem Befehl ein Unterprogramm aufgerufen werden kann. D. h. mehrere, im Hauptprogramm häufiger vorkommende Anweisungen können durchgeführt werden, bevor der Programmablauf mit dem Befehl RETURN wieder ins Hauptprogramm zurückkehrt. Das Hauptprogramm wird dann mit dem der GOSUB-Anweisung folgenden Befehl fortgesetzt. **Beispiel 3** ist ein Programm, das mit Unterprogrammaufruf arbeitet.

### 3. ON-GOTO- und ON-GOSUB-Anweisung

Diese Befehle haben folgenden Syntax:
ON var GOTO zeile1, zeile2, zeile3,...
ON var GOSUB zeile1, zeile2, zeile3,...
Hierbei wird in Abhängigkeit vom Wert der Variablen „var" zu der entsprechenden Zeile gesprungen. Hat „var" den Wert 1, wird zu „zeile 1" verzweigt, hat „var" den Wert 2, zu „zeile 2" usw. Ist „var" = 0, so wird das Programm mit der nächsten Anweisung fortgesetzt. Das gleiche geschieht auch dann, wenn der Wert von „var" größer ist als die Anzahl der angegebenen Zeilennummern. Der Ablauf der ON-GOSUB-Anweisung verläuft genauso, jedoch kehrt man aus dem Unterprogramm mit dem Befehl RETURN wieder in das aufrufende Programm zurück.
Beim **Beispiel 4** wird nach Eingabe von 1, 2 oder 3 zu der entsprechenden Pro-

grammzeile verzweigt, es sei denn, man über- oder unterschreitet den Bereich. In diesem Fall verzweigt das Programm – wie bereits beschrieben – nicht, der Programmablauf wird mit Zeile 40 fortgesetzt.

### 4. IF-THEN-GOTO (IF-GOTO-) und IF-THEN-GOSUB-Anweisung

Diese Anweisungen haben folgenden Syntax:
IF aussage THEN GOTO zeile (oder IF aussage GOTO zeile)
IF aussage THEN GOSUB zeile
Dieser Befehl läßt eine Verzweigung nur dann zu, wenn die in der Aussage formulierte Bedingung zutrifft. Der Programmablauf von **Beispiel 5** verzweigt aus Zeile 50 zur Zeile 70, wenn A den Wert 3 erreicht hat. Ist A kleiner *oder* größer als 3, wird der Programmablauf in Zeile 60 fortgesetzt. Für die IF-THEN GOSUB-Anweisung gilt analog das im 2. Abschnitt Gesagte.

In der Aussage können Bedingungen für Texte, Zahlen und Variablen gestellt werden. Zulässig sind folgende Vergleichsoperationen:

| Syntax | Bedeutung |
|---|---|
| > | größer als |
| > = oder = > | größer als oder gleich |
| < | kleiner als |
| < = oder = < | kleiner als oder gleich |
| = | gleich |
| < > oder > < | ungleich |

Der weitere Programmablauf nach einem derartigen Vergleich ist davon abhängig, ob die Bedingung zutreffend oder „wahr" ist. Einer „wahren" Bedingung wird der Zahlenwert „1" zugeordnet. Analog erhält ein „unwahrer" Vergleich den Wert „0". Zur Verdeutlichung gibt man am besten den Befehl PRINT (3 < 6) ein. Als Ergebnis wird eine „1" angezeigt, d. h. der Vergleich ist richtig (wahr). Umgekehrt ergibt die Anweisung PRINT (3 > 6) eine „0"; diese Bedingung ist unwahr. Ergebnisse von Vergleichen können auch Variablen zugeordnet werden, z. B. X = (3 < 6). Durch diese Zuweisung von „0" oder „1" als Ergebnis eines Vergleichs wird es nun möglich, verschiedene Bedingungen miteinander zu verknüpfen. Dies geschieht mit Hilfe der Boole'schen Algebra. Dabei werden die Operatoren AND, OR und NOT verwendet. Zur Erklärung wird hier kurz in Tabellenform gezeigt, welche Ergebnisse unter welchen Voraussetzungen mit den einzelnen Operatoren erreicht werden:

| Wert X Y | Operation | Ergebnis |
|---|---|---|
| 1 1 | AND | 1 |
| 1 0 | AND | 0 |
| 0 1 | AND | 0 |
| 0 0 | AND | 0 |
| 1 1 | OR | 1 |
| 1 0 | OR | 1 |
| 0 1 | OR | 1 |
| 0 0 | OR | 0 |
| 0 0 | NOT | 1 |
| 1 1 | NOT | 0 |

Mit diesen Boole'schen Operationen können nun innerhalb einer IF-THEN-Anweisung zwei oder mehrere Vergleiche miteinander verknüpft werden. So läßt zum Beispiel der Befehl IF A > B AND T$ = X$ GOTO 20 ein Programm zur Zeile 20 verzweigen, wenn
1. der Wert von A größer ist als der Wert von B und zusätzlich
2. die Stringvariable T$ gleich dem String X$ ist.

Wie schon erwähnt, können in einer Programmzeile mehrere Anweisungen stehen, die bei Erreichen der Zeile nacheinander durchgeführt werden. In der Programmiersprache Applesoft-Basic muß jedoch darauf geachtet werden, daß die Befehle, die im Anschluß an eine IF-THEN-Anweisung stehen, nur dann ausgeführt werden, wenn der Vergleich zutrifft.
Im **Beispiel 6** wird, wenn A = 3 ist, der Text „A = 3! ES WIRD VERZWEIGT!" ausgedruckt, und dann erfolgt die Verzweigung zu Zeile 70.

In der Programmiersprache Integer-Basic (der zweiten „Hochsprache" des APPLE II) sieht der Programmlauf anders aus: Wird die Zeile 50 zum ersten Mal (mit A = 1) erreicht, so trifft die Bedingung zwar nicht zu, jedoch wird trotzdem zur Zeile 70 verzweigt!
Dieser Programmablauf läßt sich in Applesoft simulieren: Dazu müssen im **Beispiel 6** die Zeilen 50 und 100 in folgender Form eingegeben werden:

```
50 ON A = 3 GOSUB 100 : GOTO 70
100 PRINT „A = 3! ES WIRD VERZWEIGT
!": RETURN
```

Nun verhält sich das Programm so, wie es bereits beschrieben wurde. Man erhält folgenden Ausdruck:

```
ZEILE 30
A = 1
ZEILE 70
```

Mit derartigen, bedingten Programmverzweigungen können nun mehrere Vergleiche in eine Programmzeile „gepackt" werden; man spart dadurch Speicherplatz.

Es folgen nun zwei Beispielprogramme (**Beispiel 7 und 8**). Das erste führt vier Verzweigungen mit IF-THEN-GOSUB durch, während das zweite die gleichen Verzweigungen mit ON-GOSUB realisiert. Der Zeitbedarf läßt sich bei beiden Programmen mit einer Stoppuhr feststellen: Beim ersten „BEEP" ist die Uhr zu starten, beim zweiten „BEEP" anzuhalten. Beide Programme zeigen selbsttätig den benötigten Speicherplatz an.
**Beispiel 7** benötigt 289 Bytes Speicherplatz und läuft ca. 9.3 Sekunden, während **Beispiel 8** 273 Bytes belegt und ca. 10 Sekunden benötigt. (Die Zeitmessung wurde, wie oben beschrieben, mit einer einfachen Stoppuhr vorgenommen.) Ein Zeitgewinn durch die Verwendung des ON-GOTO- oder des ON-GOSUB-Befehls ist demnach nicht möglich, jedoch sollte hier auch nur gezeigt werden, daß im Gegensatz zur IF-THEN-Anweisung bei der ON-GOTO-Anweisung die Möglichkeit besteht, in der gleichen Programmzeile Befehle auch dann durchführen zu lassen, wenn die vorangegangene Bedingung nicht erfüllt ist. Damit läßt sich nun in Applesoft die IF-THEN-ELSE-Anweisung simulieren.
Allerdings muß man dabei die Einschränkung in Kauf nehmen, daß bei erfüllter Bedingung zu einem anderen Programmteil verzweigt werden muß. Die Anweisung
IF A = 1 THEN GOTO 100 ELSE PRINT „FALSCH"
wird in APPLESOFT wie folgt eingegeben:
ON A = 1 GOTO 100 : PRINT „FALSCH"

Die Simulation einer Anweisung der Form
IF A = 1 THEN PRINT „RICHTIG" ELSE PRINT „FALSCH"
ist dagegen nur mit erheblich mehr Aufwand möglich.

Es empfiehlt sich, die Anwendung dieser IF-THEN-ELSE-Simulation auf die Verzweigungen nach einem umfangreichen Auswahlmenü zu beschränken, da diese Programmteile in fast allen Fällen nicht zeitkritisch sind, jedoch erheblich mehr Speicherplatz benötigen, wenn mit der IF-GOTO- oder der IF-THEN GOSUB-Anweisung verzweigt wird.

Beispiel 1

```
20  HOME
30  PRINT "ZEILE 30"
40  PRINT "ZEILE 40"
50  PRINT "ZEILE 50"
60  END
```

Beispiel 2

```
11  REM VERZWEIGUNG MIT GOTO
20  HOME
30  PRINT "ZEILE 30"
35  GOTO 50
40  PRINT "ZEILE 40"
50  PRINT "ZEILE 50"
60  END
```

Beispiel 3

```
 11  REM UNTERPROGRAMMAUFRUF
 20  HOME
 30  PRINT "ZEILE 30"
 35  GOSUB 100: PRINT "ZURUECK IM HAUPTPROGRAMM"
 40  PRINT "ZEILE 40"
 50  PRINT "ZEILE 50"
 60  END
100  PRINT "UNTERPROGRAMM"
110  PRINT "ZEILE 110": FOR I = 1 TO 500: NEXT I: REM
     KLEINE PAUSE
120  RETURN
```

Beispiel 4

```
 0  GOTO 10
 1  PRINT "ZEILE 1": GOTO 20
 2  PRINT "ZEILE 2": GOTO 20
 3  PRINT "ZEILE 3": GOTO 20
11  REM VARIABLE VERZWEIGUNG
15  HOME
20  VTAB 10: PRINT "ZEILE (1-3) ";: GET A%
25  PRINT A%: VTAB 12
30  ON A% GOTO 1,2,3
40  PRINT "PROGRAMMENDE"
50  END
```

Beispiel 5

```
 6  GOTO 30
11  REM BEDINGTE VERZWEIGUNG
20  HOME :A = 0
30  PRINT "ZEILE 30"
40 A = A + 1
50  IF A = 3 THEN  GOTO 70
60  GOTO 30
70  PRINT "ZEILE 70"
80  END
```

Beispiel 6

```
 10  HOME
 20 A = 0
 30  PRINT "ZEILE 30"
 40 A = A + 1: PRINT "A="A
 50  IF A = 3 THEN  GOSUB 100: GOTO 70
 60  GOTO 30
 70  PRINT "ZEILE 70"
 80  END
100  PRINT "A=3 ! ES WIRD VERZWEIGT !": FOR I = 1 TO 500:
     NEXT I: RETURN
```

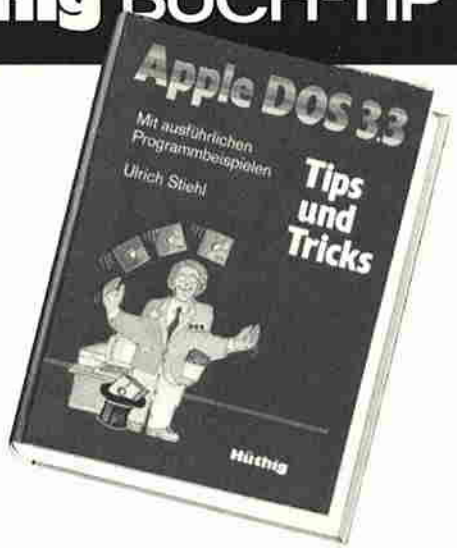
Beispiel 7

```
  10  REM
  20  REM SPEICHERPLATZTEST
  30  REM
  50  INVERSE : FOR I = 1 TO 5: GOSUB 1010: NEXT : PRINT
      CHR$ (7)" START ": NORMAL
 100  FOR I = 1 TO 5: FOR J = 1 TO 100
 120  IF J = 1 THEN  GOSUB 1000
 130  IF J = 2 THEN  GOSUB 1000
 140  IF J = 3 THEN  GOSUB 1000
 150  IF J = 4 THEN  GOSUB 1000
 160  NEXT J: NEXT I
 170  PRINT : PRINT "SPEICHERPLATZ : " PEEK (106) * 256 +
      PEEK (105) - ( PEEK (104) * 256 +  PEEK (103)) - 1
 180  INVERSE : PRINT  CHR$ (7)" ENDE ": GOSUB 1010: NORMAL
 190  END
1000  PRINT J
1010  FOR K = 1 TO 100: NEXT K: RETURN
```

Beispiel 8

```
  10  REM
  20  REM SPEICHERPLATZTEST
  30  REM
  50  INVERSE : FOR I = 1 TO 5: GOSUB 1010: NEXT : PRINT
      CHR$ (7)" START ": NORMAL
 100  FOR I = 1 TO 5: FOR J = 1 TO 100
 120  ON J = 1 GOSUB 1000: ON J = 2 GOSUB 1000: ON J = 3
      GOSUB 1000: ON J = 4 GOSUB 1000
 160  NEXT J: NEXT I
 170  PRINT : PRINT "SPEICHERPLATZ : " PEEK (106) * 256 +
      PEEK (105) - ( PEEK (104) * 256 +  PEEK (103)) - 1
 180  INVERSE : PRINT  CHR$ (7)" ENDE ": GOSUB 1010: NORMAL
 190  END
1000  PRINT J
1010  FOR K = 1 TO 100: NEXT K: RETURN
```


**Hüthig BUCH-TIP**




**Apple DOS 3.3 — Tips und Tricks**

**von U. Stiehl**

2. Aufl. 1984, 216 S., mit zahlreichen, ausführlich kommentierten Programmlistings, kart., DM 28,—
ISBN 3-7785-1049-5

**Dr. Alfred Hüthig Verlag · Postf. 10 28 69 · 6900 Heidelberg 1**

# Wie man Arrays löscht

**von Christiane Breit**
**und Thomas Schneyer**

## Wozu Arrays löschen?

Die bei der Programmierung in Applesoft mögliche Verwendung von Arrays (d. h. dimensionierten Variablen) weiß wohl jeder zu schätzen, der eine größere Menge von Daten zu bearbeiten hat, die zu einer gemeinsamen Kategorie gehören. Hierfür ein aktuelles Beispiel:

Im Rahmen einer wissenschaftlichen Untersuchung über die Auftretenshäufigkeit bestimmter Krankheiten in Abhängigkeit von der Umweltbelastung des Wohnortes sollen pro Arztpraxis 200 Krankheitsbilder in einem Textfile erfaßt werden. Für die weitere statistische Verarbeitung der Daten ist man nur an den 20 häufigsten Symptomen interessiert.

Die Bezeichnungen der Krankheiten werden in einem Array DA$ (200), die dazugehörigen Häufigkeiten im Array DA (200) gespeichert. Anschließend wird nach der Häufigkeit so sortiert, daß die 20 häufigsten Krankheiten in den Feldern des Arrays DS$ (20) stehen. Im weiteren Programm werden DA$ (200) und DA (200) nicht mehr benötigt und es wäre wünschenswert, über den für sie reservierten Speicherplatz für andere Zwecke verfügen zu können. In Applesoft gibt es jedoch kein Kommando, das das Löschen einzelner Arrays gestattet. Diese Aufgabe erfüllt das in diesem Artikel vorgestellte Unterprogramm.

## Wo befinden sich Arrays im Speicher?

Informationen über Arrays, die das Programm zur Verarbeitung benötigt, werden in der sog. Array-Table (ARYTAB) festgehalten. Die Array-Table ist ein Bereich des Speichers, der im Normalfall hinter den Speicherbereichen für das Applesoft-Programm und für die einfachen Variablen liegt. Informationen über diese internen Speicheraufteilungen werden von Applesoft in Adressen der Zero-Page abgelegt. Diese Speicherplätze werden Pointer oder Zeiger genannt, da ihre Inhalte oft Anfang oder Ende von speziellen Speicherabschnitten angeben. Sie bestehen in der Regel aus 2 Bytes, wobei das erste das Low Byte und das zweite das High Byte der entsprechenden Adresse enthält („6B 00" ist die Adresse $006B).

Im Falle der dimensionierten Variablen sind zwei Pointer von Bedeutung: Der Array-Table-Pointer (ARYTAB) in Adresse $006B-$006C bzw. dezimal 107-108 enthält die Anfangsadresse des Bereiches, in dem Arrays abgespeichert werden.

Im Storage-End-Pointer (STREND) in Adresse $006D-$006E bzw. 109-110 befindet sich die dazugehörige Endadresse + 1.

Der Inhalt der Pointer ARYTAB und STREND ändert sich bei jeder ins Programm neu aufgenommenen einfachen Variablen, jedem neuen Array und bei jeder Modifikation, die die Länge des Programms beeinflußt, d. h. die Array-Table verschiebt sich.

## Wie werden Arrays intern abgespeichert?

Die Array-Table besteht aus reservierten Bereichen (Variable-Table, VARTAB) für die einzelnen dimensionierten Variablen, die Informationen über Variablentyp (REAL, INTEGER, STRING), Dimensionsanzahl, Länge und Wert enthalten. Je nach Anzahl und Umfang der in Applesoftprogrammen festgelegten Arrayfelder variiert deren Größe. Der Aufbau der drei verschiedenen Variable-Table-Typen wird im folgenden anhand des Beispiels beschrieben.

Die Eingabe folgender Programmzeilen liefert als Ergebnis den Hexdump aus **Beispiel 1**

```
NEW
CALL -151 (Sprung in Monitor)
*0800: 00
*0801<0800.1000M (löscht Speicher)
Ctrl-C
100 DIM DA (2), DA$ (4), D% (2, 4)
RUN
CALL -151 (Sprung in Monitor)
*6B.6C (Abfrage ARYTAB)
006B- 1E 08 (Anfangsadresse Array-
Table)
*6D.6E (Abfrage STREND)
006D- 71 08 (Endadresse Array-Table +
1)
*0818.0877 (Hexdump: Beispiel 1)
```

Der Beginn der Variable-Table für DA (2) ist bei $081E, für DA$ (4) bei $0834 und für D% (2, 4) bei $084A.

**Byte 1 und 2** jeder Variable-Table enthalten den Namen der dimensionierten Variablen im ASCII-Code. Besteht der Variablenname nur aus einem Buchstaben, so erhält das 2. Byte den Wert 0. Zum ASCII-Wert des 2. Bytes wird bei Variablen des Typs String und Integer $80 addiert, bei Integer-Variablen ebenso zum 1. Byte. Die Variablenbezeichnung DA wird daher als „44 41" codiert, DA$ als „44 C1" und D% als „C4 80".

**Byte 3 und 4** (Low-High) der jeweiligen Variable-Table enthalten den sog. Offsetpointer. Sein Wert gibt an, wieviele Speicherplätze vom Low-High-Beginn der Variable-Table entfernt der Anfang der nächstfolgenden Variable-Table liegt (im Beispiel 1: „16 00", „16 00" und „27 00").

**Byte 5** gibt die Anzahl der Dimensionen an („01", „01" und „02").

**Byte 6 und 7** (High-Low) enthalten die Größe der am weitesten rechts angegebenen Dimension („00 03", „00 05" und „00 05"), d. h. die „4" bei D% (2, 4) ist am weitesten rechts. Die Werte sind um 1 größer als im Dimensionsbefehl angegeben, da bei Applesoft eine 0. Dimension automatisch vorgesehen ist. Jede weitere angegebene Dimension beansprucht zu-

sätzliche 2 Bytes zur Angabe ihrer Größe. Im Beispiel 1 trifft dies lediglich für D% (2, 4) zu. Byte 8 und 9 enthalten daher „00 03".

Daran schließen sich Datenwertfelder bzw. Pointerfelder an. Bei Real werden für jeden Datenwert 5 Bytes, bei Integer 2 Bytes reserviert. Im Beispiel bei DA (2) stehen ab dem 8. Byte 3 Felder zu je 5 Bytes, bei D% (2,4) ab dem 10. Byte 3 mal 5 = 15 Felder zu je 2 Bytes.
Bei Strings stehen die eigentlichen Daten nicht in der Variable-Table, da durch die sehr unterschiedliche Länge notgedrungen pro Arrayfeld 255 Bytes reserviert werden müßten, was bei Strings der Länge 2 eine Platzverschwendung von 99% wäre. Im Variablenspeicher sind daher pro Stringfeld nur drei Bytes reserviert. Im ersten steht die Länge, d.h. die Anzahl der Zeichen, im zweiten und dritten die Adresse (Low-High), ab welcher der String abgespeichert ist. Im Beispiel 1 sind dies 5 Felder zu je 3 Bytes.

## Was muß das Unterprogramm „Löschen eines Arrays" leisten?

Die Aufgabe dieses Unterprogramms besteht darin, den reservierten Platz für einen zuvor dimensionierten Array wieder freizugeben. Zu diesem Zweck sind folgende Schritte erforderlich:

1. Erkennen des Typs des zu löschenden Arrays
2. Finden des Arrays
3. Berechnen des Offset zum nachfolgenden Array
4. Verschieben der nachfolgenden Arrays
5. Freigeben des gewonnenen Speicherplatzes

### Erläuterungen zum Programm 1

*Benutzte Variablen*

A$ = Name und Typ des zu löschenden Arrays (wird vom Hauptprogramm übergeben)
A = ASCII-Wert des 1. Buchstabens dieses Arraynamens
A1 = ASCII-Wert des 2. Buchstabens dieses Arraynamens
(Je nach Arraytyp werden bei A und A1 128 = $80 hinzugezählt.)
K = Schleifenvariable
WA = Anfang einer Variable-Table
WE = Ende der Array-Table + 1
WF = Offset zur nächsten Variable-Table
W1 = Kennung für Arraytyp

*Zum Programmablauf*

Im Programm sind alle Zahlen im Dezimalsystem angegeben. In Zeile 40100 werden alle im Unterprogramm benutzten Variablen auf den Wert 0 gesetzt. Dies ist notwendig, da eine später im Programm hinzukommende Variable eine Verschiebung des STREND und möglicherweise auch des ARYTAB bewirken und somit WE und WA falsche Werte annehmen würden. In Zeile 40110 wird zur Feststellung des Arraytyps das rechte Zeichen von A$ geprüft. „ASC (RIGHT$ (A$,1) < 38" ist ein logischer Ausdruck, der die Werte 0 = falsch oder 1 = wahr annimmt. Er wird dann 1, wenn der ASCII-Wert des rechten Zeichens von A$ kleiner als 38 ist. W1 erhält den Wert 0 beim REAL-Array, 1 beim STRING-Array und 2 beim Integer-Array. In Zeile 40115 erhält A den ASCII-Wert des 1. Namensbuchstaben + 128, falls es sich um einen Integer-Array handelt. In den Zeilen 40120 und 40125 geschieht dasselbe für den eventuellen 2. Buchstaben.
In den Zeilen 40130 und 40135 werden ARYTAB und STREND eingelesen. Falls Anfangs- und Endadresse der Table identisch sind – d. h. es gibt keine Arrays – wird von Zeile 40140 aus auf „RETURN" gesprungen. Die Schleife von 40150–40160 wird so lange durchlaufen, bis entweder der Array gefunden (K bleibt 1) oder das Ende der Array-Table erreicht ist (K wird 1 gesetzt). In 40153 wird der Variablen WF der Offset zur nächsten Variable-Table zugewiesen. Falls der Arrayname gefunden wurde, wird zu „NEXT" gesprungen und somit die Schleife verlassen. In Zeile 40156 erhält WA den Wert der Anfangsadresse der nächsten Variable-Table. Falls die Array-Table noch nicht vollständig abgesucht ist, wird in 40157 die Schleifenvariable K wieder auf 0 gesetzt. Falls das Array-Table-Ende erreicht wurde, erfolgt in Zeile 40162 ein Sprung nach „RETURN". In Zeile 40164 wird das ursprüngliche Ende der Array-Table um die Länge des gelöschten Arrays zurückgesetzt. Von 40170–40175 werden alle der gelöschten Variable-Table folgenden Tables um den nun freigewordenen Bereich nach vorne verschoben. In den Zeilen 40180–40185 erhält STREND den aktuellen Wert.

Die vorangestellten Zeilen 200–999 dienen – falls noch kein eigenes Hauptprogramm vorhanden ist – zum kurzen Testen der Funktionsweise dieses Unterprogramms. Es werden Arrays verschiedenen Typs dimensioniert und einer davon zum Löschen markiert. Durch Vergleich der Array-Table entsprechend oben beschriebener Befehlsfolge (Beispiel 1) kann man sich nach Änderung von Zeile 249 vom Funktionieren dieser Technik überzeugen.

Bei Verzicht auf Überprüfen falscher Arraynamen oder durch direkte Übergabe des Variablennamens läßt sich das Unterprogramm beträchtlich verkürzen.

Außer dem der weiteren Datenverarbeitung wieder zugänglichen Speicherplatz bewirkt das Unterprogramm eine Beschleunigung des Programmablaufs, der besonders bei großer Anzahl von Stringvariablen ins Gewicht fällt. Die Garbage-Collection (das Aufräumen des nicht mehr benötigten Stringplatzes) findet seltener statt.

```
Beispiel 1

0818- 00 00 00 00 00 00 44 41
0820- 16 00 01 00 03 00 00 00
0828- 00 00 00 00 00 00 00 00
0830- 00 00 00 00 44 C1 16 00
0838- 01 00 05 00 00 00 00 00
0840- 00 00 00 00 00 00 00 00
0848- 00 00 C4 80 27 00 02 00
0850- 05 00 03 00 00 00 00 00
0858- 00 00 00 00 00 00 00 00
0860- 00 00 00 00 00 00 00 00
0868- 00 00 00 00 00 00 00 00
0870- 00 FF 00 00 00 00 00 00
```

```
Programm 1

199 REM Testprogramm zu Unterprogramm
    "Löschen eines Arrays"
200 DIM DA(2),DA$(4),D%(2,4)
240 D%(2,4) = 256
249 A$ = "DA$": REM des zu löschenden Arrays
250 GOSUB 40100
300 PRINT D%(2,4): REM zeigt, daß der Variablenwert
    noch vorhanden ist
350 DIM DA$(200)
999 END


40099 REM Unterprogramm: Löschen eines Arrays
40100 A = 0: A1 = 0: K = 0: WA = 0:
      WE = 0: WF = 0: W1 = 0
40110 W1 = (ASC (RIGHT$ (A$,1)) < 38) +
      (ASC (RIGHT$ (A$,1)) = 37)
40115 A = ASC ( LEFT$ (A$,1)) + 128 * (W1 = 2)
40120 IF (LEN (A$) > 1) THEN A1 =
      (ASC (MID$ (A$,2,1)) > 64) * ASC(MID$(A$,2,1)
40125 A1 = A1 + (W1 > 0) * 128
40130 WA = PEEK (107) + 256 * PEEK (108)
40135 WE = PEEK (109) + 256 * PEEK (110)
40140 IF WA = WE THEN GOTO 40199
40150 FOR K = 1 TO 1
40153 WF = PEEK (WA + 2) + 256 * (PEEK (WA + 3))
40155 IF (PEEK (WA) = A AND PEEK (WA + 1) = A1)
      THEN GOTO 40160
40156 WA = WA + WF
40157 K = (WA > = WE)
40160 NEXT
40162 IF (WA > = WE) THEN GOTO 40199
40164 WE = WE - WF
40170 FOR K = WA TO WE
40173 POKE K, PEEK (K + WF)
40175 NEXT
40180 POKE 110, WE/256
40185 POKE 109, WE - INT(WE / 256) * 256
40199 RETURN
```

| | | | | | | |
|---|---|---|---|---|---|---|
| | $0871 | FF | | | | Ende + 1 der Array-Table |
| VARTAB 3. dimensionierte Variable | $086F | 00 | 00 | | | Wert von D% (2,4) |
| | $086D | 00 | 00 | | | " D% (1,4) |
| | $086B | 00 | 00 | | | " D% (0,4) |
| | $0869 | 00 | 00 | | | " D% (2,3) |
| | $0867 | 00 | 00 | | | " D% (1,3) |
| | $0865 | 00 | 00 | | | " D% (0,3) |
| | $0863 | 00 | 00 | | | " D% (2,2) |
| | $0861 | 00 | 00 | | | " D% (1,2) |
| | $085F | 00 | 00 | | | " D% (0,2) |
| | $085D | 00 | 00 | | | " D% (2,1) |
| | $085B | 00 | 00 | | | " D% (1,1) |
| | $0859 | 00 | 00 | | | " D% (0,1) |
| | $0857 | 00 | 00 | | | " D% (2,0) |
| | $0855 | 00 | 00 | | | " D% (1,0) |
| | $0853 | 00 | 00 | | | D% (0,0) |
| | $0851 | 00 | 03 | | | Größe der 1. Dimension |
| | $084F | 00 | 05 | | | Größe der 2. Dimension |
| | $084E | 02 | | | | Anzahl der Dimensionen |
| | $084C | 27 | 00 | | $084A + $0027 = $0871 | Beginn der nächsten VARTAB |
| | $084A | C4 | 80 | | D% | Name und Variablentyp |
| VARTAB 2. dimensionierte Variable | $0847 | 00 | 00 | 00 | 1. Byte Stringlänge, 2. + 3. Byte Stringebene | DA$ (4) |
| | $0844 | 00 | 00 | 00 | " " | DA$ (3) |
| | $0841 | 00 | 00 | 00 | " " | DA$ (2) |
| | $083E | 00 | 00 | 00 | " " | DA$ (1) |
| | $083B | 00 | 00 | 00 | " " | DA$ (0) |
| | $0839 | 00 | 05 | | | Größe der Dimensionen |
| | $0838 | 01 | | | | Anzahl der Dimensionen |
| | $0836 | 16 | 00 | | $0834 + $0016 = $084A | Beginn der nächsten VARTAB |
| | $0834 | 44 | C1 | | DA$ | Name und Variablentyp |
| VARTAB 1. dimensionierte Variable | $082F | 00 | 00 | 00 00 00 | | Wert von DA (2) |
| | $082A | 00 | 00 | 00 00 00 | | " DA (1) |
| | $0825 | 00 | 00 | 00 00 00 | | " DA (0) |
| | $0823 | 00 | 03 | | | Größe der Dimensionen |
| | $0822 | 01 | | | | Anzahl der Dimensionen |
| | $0820 | 16 | 00 | | $081E + $0016 = $0834 | Beginn der nächsten VARTAB |
| | $081E | 44 | 41 | | DA | Name und Variablentyp |
| | ⋮ | | | | | Beginn der Array -Table |
| | $006D | 71 | 08 | | STREND | Ende der Array-Table + 1 |
| | $006B | 1E | 08 | | ARYTAB | Pointer für Beginn der Array-Table |
| | ⋮ | | | | | |


# DB-MEISTER

**Adreß- und Schemabriefprogramm**

*Der DB-Meister ist ein in Assembler geschriebenes, ungewöhnlich schnelles, unkompliziertes und zugleich „narrensicheres" Adreß-, Datei- und Schemabriefprogramm.*

*Der DB-Meister dient zum Anlegen, Pflegen, Sortieren, Selektieren und Ausdrucken von Dateien aller Art. Als Apple-Benutzer wissen Sie, wie langsam viele Programme dieser Art sind. Nicht so der DB-Meister!*

Drei Beispiele:

- Jeder beliebige von 560–999 Records wird nach Indexfeldern in 0,2 Sekunden gefunden.
- Eine komplette Datendiskette mit z. B. 600 Records läßt sich in 1 Minute nach 3 Feldern sortieren und untersortieren. Dabei ist die Zeit für Diskettenzugriff bereits mitgerechnet.
- Das Einlesen eines 50 Sektoren langen Programm-Moduls dauert nur 3,5 Sekunden.

*Technische Daten des DB-Meisters*

- Recordlänge bis zu 230 Zeichen
- 560 bis 1000 Records pro Datendiskette
- Maximal 25 Felder pro Record
- 4 Datentypen (String, Integer, Dezimalzahl, Real)
- Suche nach 3 Indexfeldern – je 4 Zeichen lang – mit Wildcard-Funktion
- Sortieren und Filtern (kumuliertes Selektieren) geschieht nach den Index-Feldern
- Ausdruck der Dateien als Etiketten, Listen und Schemabriefe (mit Felder- und Tastatureinschüben an beliebigen Stellen des Formbriefes)
- normal kopierbare Programmdiskette, unterteilt in Hauptprogramme und diverse Hilfsprogramme
- einsatzfähig auf Apple IIe oder IIc. (Achtung: Brief-Modul läuft nicht mit Videx-Karte!)
- 256K RAM-Disks verwendbar

**Gesamtpreis 290,– (2 Disketten + gedrucktes Manual)**
**U. Stiehl**
**c/o Dr. A. Hüthig Verlag**
**Postfach 10 28 69 · 6900 Heidelberg**

File Edit Control
Peeker Demo
List
10 PRINT "Dies ist mein erstes"
20 PRINT "Programm in
30 PRINT "Microsoft-Basic"
40 PRINT: PRINT
50 PRINT "A MECKI A DAY"
60 PRINT "KEEPS THE DOLDRUMS AWAY"
70 PRINT: PRINT: PRINT
Dies ist mein erstes
Programm in
Microsoft-Basic
A MECKI A DAY
KEEPS THE DOLDRUMS AWAY
Command
LIST

# Micro

**In dieser mehrteiligen Artikelserie wird das Microsoft-Basic Version 1.00.00 für den Macintosh systematisch und von Grund auf für Basic-Programmierer dargestellt. Deshalb sollte der Leser bereits mit einem Basic-Dialekt, z.B. Applesoft-Basic, auf das vergleichend hingewiesen wird, sowie darüber hinaus mit der allgemeinen Macintosh-Bedienung hinsichtlich Fensterverschiebung usw. vertraut sein.**
**Der erste Teil umfaßt neben einer Einführung in die Fenster und Menüs das zunächst noch etwas trockene, aber unentbehrliche definitorische Grundgerüst der Operatoren und Operanden. In den nachfolgenden Teilen werden dann die Funktionen, Programmsteuerbefehle, Ein- und Ausgabeoperationen sowie die Grafik- und ROM-Routinen anhand konkreter Beispiele erläutert.**

# soft Basic leicht geMACht

von Pit Capitain

## Teil 1: Operanden und Operatoren

Im Lieferumfang des Microsoft-Basic ist eine 3 1/2 Zoll Diskette mit dem Basic-System und einigen Beispielprogrammen sowie ein Handbuch in englischer Sprache enthalten. Das Handbuch umfaßt 215 Seiten. Es beginnt mit einer Einführung in das Arbeiten mit dem Microsoft-Basic. Anschließend werden die speziellen Eigenschaften dieses Basic-Dialekts vorgestellt. Es folgt ein ausführlicher Referenz-Teil, in dem die Basic-Befehle alphabetisch geordnet erklärt werden. Bei jedem Befehl ist seine Syntax, seine Bedeutung, ein kurzes Programm-Beispiel und manchmal eine Warnung oder ein sonstiger Hinweis angegeben. Im Anhang befindet sich u.a. eine Liste mit den von Basic aus erreichbaren ROM-Routinen, allerdings ohne nähere Beschreibung, sowie zwei kleinere Beispielprogramme.
Um das Basic-System zu starten, muß man das Programm „MS-BASIC" wie üblich öffnen (z.B. mit einem Doppelklick der Maus). Am Anfang sieht der Schreibtisch (also der Bildschirm) wie in Abb. 1 aus. Es ist das Ausgabefenster und das aktive Befehlsfenster geöffnet. Die einzelnen Fenster werden im folgenden beschrieben.

### 1. FENSTER UND MENÜS

#### 1.1. Fenster

Beim Microsoft-Basic gibt es drei verschiedene Arten von Fenstern: das Befehlsfenster, das Ausgabefenster und ein oder mehrere Listfenster. Alle Fenster können in ihrer Lage auf dem Schreibtisch und in ihrer Größe verändert werden, indem man die üblichen Aktionen mit der Maus ausführt.

*1.1.1. Befehlsfenster*

Das Befehlsfenster ist das Fenster mit dem Namen „Command" (vgl. Abb. 1). Im Befehlsfenster erscheinen alle Befehle, die der Benutzer mit der Tastatur eingibt. Dazu gehören sowohl die direkte Eingabe von Basic-Befehlen als auch die Eingabe und Änderung von Programmzeilen. Gibt man z.B. eine Programmzeile ein, so erscheinen die gedrückten Tasten im Befehlsfenster. Wenn die RETURN-Taste betätigt wird, dann wird diese Zeile in den Speicher übernommen. Es werden im Gegensatz zum Applesoft-Basic keine Leerzeichen ergänzt. Dadurch sind z.B. Einrückungen von Zeilen möglich. Jede Programmzeile kann übrigens aus bis zu 255 Zeichen bestehen. Die Zeilennummern müssen im Bereich von 0 bis 65529 liegen. (Beim Applesoft-Basic: bis zu 239 Zeichen pro Zeile, Zeilennummern von 0 bis 63999.)
Das Befehlsfenster befindet sich immer auf dem Schreibtisch, wenn gerade kein Basic-Programm läuft. Sobald Tasten beim Macintosh gedrückt werden, wird das Befehlsfenster zum aktiven Fenster (das Fenster, mit dem gerade gearbeitet werden kann).

*1.1.2. Ausgabefenster*

Dieses Fenster entspricht (fast) dem normalen Bildschirm beim Applesoft-Basic. Alle Eingaben aus dem Befehlsfenster werden im Ausgabefenster wiederholt, sobald sie mit der RETURN-Taste abgeschlossen werden (z.B. in Abb. 2 das Wort „RUN"). Außerdem erscheinen alle Ausgaben von Basic-Programmen im Ausgabefenster. Dort wird auch die Antwort des Benutzers auf Basic-Befehle wie „INPUT" etc. ausgegeben.

Das Ausgabefenster hat denselben Namen wie das Basic-Programm, das sich gerade im Speicher befindet. Wenn kein Basic-Programm geladen ist, ist der Name des Ausgabefensters „Untitled" (s. Abb. 1 und 2).

Die Länge der Zeilen, die im Ausgabefenster ausgegeben werden, läßt sich mit dem Basic-Befehl „WIDTH" ändern (wird in einer der nächsten Folgen erklärt). Am Anfang ist die Zeilenlänge des Ausgabefensters „unendlich", das heißt, daß bei der Ausgabe nirgends innerhalb von Zeilen ein Wagenrücklauf (CR) eingefügt wird, auch wenn die Zeilen viel länger als das Ausgabefenster sind.

Ein weiterer Unterschied zwischen dem Bildschirm beim Applesoft-Basic und dem Ausgabefenster besteht darin, daß Programmlistings nicht im Ausgabefenster er-

scheinen. Dafür gibt es eine besondere Fenstersorte, die Listfenster.

*1.1.3. Listfenster*

Das Fenster in Abb. 2 mit dem Namen „List" ist ein Listfenster. Listfenster werden dazu gebraucht, um den Text eines Basic-Programms auszugeben. Der Benutzer kann ein, zwei oder auch drei Listfenster öffnen. Jedes Listfenster kann dann einen anderen Teil des Programms darstellen. Ein Listfenster zeigt normalerweise nur einen Ausschnitt des Basic-Programms, das sich gerade im Speicher befindet. Man kann den gezeigten Ausschnitt an jede Stelle des Textes bewegen, indem man mit der Maus wie üblich die „Scroll"-Kontrollen am rechten und am unteren Rand des aktiven Listfensters betätigt.
Ein neues Listfenster wird durch den (Menü- oder Basic-)Befehl „LIST" geöffnet (s. 1.2.3.). Der Programmtext im Listfenster wird beim Editieren des Textes automatisch auf den neuesten Stand gebracht. Jedes Listfenster zeigt also immer den aktuellen Zustand. Ein aktives Listfenster wird geschlossen, indem man wie üblich die Maus in dem kleinen, weißen Quadrat in der oberen, linken Ecke des Listfensters drückt (vgl. Abb. 2).
Der Benutzer kann ein Listfenster aktivieren, indem er den Cursor mit der Maus irgendwo in das Listfenster bringt und den Knopf der Maus drückt. Wird die Maus in einem Listfenster gedrückt, das schon aktiviert ist, dann wird die Zeile, auf die der Cursor zeigt, in das Befehlsfenster kopiert. Dort kann sie dann editiert werden. Wird danach auf die RETURN-Taste gedrückt, wird die geänderte Zeile in den Speicher übernommen. Das oder die Listfenster werden dann entsprechend korrigiert.
Beispiele von Listfenstern sieht man in den Listings der Beispielprogramme (z.B. Abb. 2).

## 1.2. Menüs

Die Menüs beim Microsoft-Basic sind aus Abb. 1 ersichtlich. Mit dem Apfel-Menü kann man auf die Schreibtisch-Utensilien wie beispielsweise Notizblock oder Taschenrechner zugreifen. Mit dem „File"-Menü lassen sich Dateien bearbeiten. Das „Edit"-Menü dient zum Editieren des Programmtextes. Das „Control"-Menü enthält verschiedene Basic-Kontrollbefehle. Die einzelnen Befehle der Menüs werden nachstehend kurz erklärt. Dazu gehört jedoch nicht das Apfel-Menü, da es unter Basic genauso funktioniert wie sonst auch.

*1.2.1. „File"-Menü*

**New**: Der Programm- und Datenspeicher wird gelöscht. Alle Listfenster werden geschlossen, das Ausgabe- und das Befehlsfenster gelöscht. Falls sich ein Basic-Programm im Speicher befindet, das noch nicht auf Diskette gespeichert ist, wird der Benutzer gefragt, ob er dies nachholen möchte. Dieser Menü-Befehl hat dieselbe Wirkung wie der Basic-Befehl „NEW".

**Open...**: Von der Diskette wird ein Programm geladen, das anschließend gestartet, aufgelistet oder editiert werden kann. Dadurch wird natürlich der Programmspeicher verändert. Falls sich also vorher noch ein Basic-Programm im Speicher befindet, das noch nicht auf Diskette gespeichert ist, wird der Benutzer gefragt, ob er dies nachholen möchte. Nachdem man den Open-Befehl gewählt hat, erscheint ein sogenanntes Dialogfenster, in welchem der Name des zu ladenden Programms eingegeben werden kann (vgl. auch Abb. 3).
**Close**: Dieser Befehl schließt normalerweise Dateien, die mit dem „Open"-Befehl geöffnet wurden. Der Menü-Befehl „Open" öffnet nur Programm-Dateien (s.o.). Diese werden aber mit dem Menü-Befehl „Save" (s.u.) geschlossen, d.h. daß der Menü-Befehl „Close" eigentlich überflüssig ist. Insofern erscheint dieser Befehl im Menü immer nur in grauer Schrift, was bedeutet, daß er zur Zeit nicht ausgeführt werden kann. Im Basic-Handbuch wird dieser Menü-Befehl auch nirgends erwähnt.

**Save** und **Save As...**: Das im Speicher befindliche Basic-Programm wird auf Diskette geschrieben. Nachdem man den Befehl „Save As..." gewählt hat, erscheint das Dialogfenster in Abb. 3, in welchem der Name der Datei und das Format, in dem das Programm gespeichert werden soll, festgelegt wird. Falls man nur die neue Version eines Programms unter dem alten Namen abspeichern will, wählt man den Befehl „Save"; dann erscheint das Dialogfenster nicht. Die drei möglichen Ausgabeformate bedeuten folgendes: ein Programm im *binären* Format ist kürzer als dasselbe Programm im *ASCII*-Format, da Basic-Befehle wie z.B. „INPUT" in gekürzter Form abgespeichert werden. Ein Programm, das in *geschütztem* Format abgespeichert wurde, ist genauso lang wie im binären Format. Es kann aber nach erneutem Laden nicht mehr aufgelistet werden.
**Quit**: Das Basic-System wird verlassen. Alle geöffneten Dateien werden geschlossen. Man befindet sich nach diesem Befehl in demselben Zustand wie vor dem Starten des Basic-Systems. Falls sich noch ein Basic-Programm im Speicher befindet, das noch nicht auf Diskette gespeichert ist, wird der Benutzer gefragt, ob er dies nachholen möchte. Dieser Menü-Befehl hat dieselbe Wirkung wie der Basic-Befehl „SYSTEM" (wird später erklärt).

*1.2.2. „Edit"-Menü*

Um Text bearbeiten zu können, gibt es beim Macintosh einen Text-Zwischenspeicher, in den ein mit der Maus ausgewählter Text eingeschrieben werden kann. Dieser Text kann später an einer anderen Stelle wieder aus dem Text-Zwischenspeicher in den Programmtext eingefügt werden (zum Text-Zwischenspeicher s. auch 1.3.). Die Befehle im „Edit"-Menü benutzen den Text-Zwischenspeicher wie folgt:

**Cut**: Der mit der Maus ausgewählte Text aus dem Befehlsfenster wird in den Text-Zwischenspeicher kopiert und danach im Befehlsfenster gelöscht. Der alte Inhalt des Text-Zwischenspeichers geht verloren.
**Copy**: Der mit der Maus ausgewählte Text aus dem Befehlsfenster wird in den Text-Zwischenspeicher kopiert. Der Text im Befehlsfenster ändert sich dadurch nicht. Der alte Inhalt des Text-Zwischenspeichers geht verloren.
**Paste**: Der Inhalt des Text-Zwischenspeichers wird bei der mit der Maus markierten Text-Position im Befehlsfenster eingefügt. Der Text-Zwischenspeicher ändert sich dadurch nicht.

*1.2.3. „Control"-Menü*

**Stop**: Ein laufendes Basic-Programm wird unterbrochen. Im Ausgabefenster erscheint die Meldung „Break in nnn" mit der augenblicklichen Zeilennummer. Das Befehlsfenster, das während des Programmlaufs geschlossen war, wird wieder geöffnet. Dieser Menü-Befehl hat dieselbe Wirkung wie der Basic-Befehl „STOP". Er kann auch über die Tastatur aufgerufen werden, indem man die „Command"-Taste gleichzeitig mit der Taste „C" drückt. Dies entspricht den Tasten „Ctrl-C" beim Applesoft-Basic.
**Continue**: Ein Programm, das durch den (Menü- oder Basic-)Befehl „STOP" unterbrochen wurde, wird an derselben Stelle fortgesetzt. Dieser Menü-Befehl hat dieselbe Wirkung wie der Applesoft-Basic-Befehl „CONT".

**Suspend**: Das gerade laufende Basic-Programm wird angehalten, bis irgendeine Taste gedrückt wird. Dieser Befehl kann auch über die Tastatur aufgerufen werden, indem man die „Command"-Taste gleichzeitig mit der Taste „S" drückt. Dies entspricht den Tasten „Ctrl-S" beim Applesoft-Basic.
**List**: Ein neues Listfenster wird geöffnet. Der im Fenster gezeigte Ausschnitt befindet sich ganz am Anfang des Programmtextes. Dieser Menü-Befehl hat dieselbe Wirkung wie der BASIC-Befehl „LIST".
**Run**: Das im Speicher befindliche Programm wird gestartet. Die Ausführung beginnt mit der ersten Programmzeile. Dieser Menü-Befehl hat dieselbe Wirkung wie der Basic-Befehl „RUN".
**Trace On**: Der „Trace-Modus" wird eingeschaltet. In diesem Zustand wird bei einem laufenden Programm jede Zeilennummer im Ausgabefenster ausgegeben, sobald die entsprechende Programmzeile ausgeführt wird. Das kann bei der Fehlersuche recht hilfreich sein. Dieser Menü-Befehl hat dieselbe Wirkung wie der Applesoft-Befehl „TRACE".
**Trace Off**: Der „Trace-Modus" wird ausgeschaltet. Dieser Menü-Befehl hat dieselbe Wirkung wie der Applesoft-Befehl „NOTRACE".

### 1.3. Dateinamen

Bei der Ein- und Ausgabe von Daten stellt sich die Frage nach den erlaubten Dateinamen. Im Microsoft-Basic kann ein Dateiname aus bis zu 255 Zeichen (keine Sonderzeichen) bestehen. Es sind sowohl Groß- als auch Kleinbuchstaben erlaubt, allerdings wird beim Öffnen einer Datei nicht z.B. zwischen den Namen „Sound-Test" und „SOUNDTEST" unterschieden. Ähnlich wie beim UCSD-Pascal kann vor dem Dateinamen der Name des angesprochenen Gerätes (der „Volume-Name") angegeben werden. Er wird durch ein „:" vom Dateinamen getrennt. Wenn man z.B. die Datei „Monitor.14" auf der Diskette mit dem Namen „MSBasic Arbeit" ansprechen will, so lautet der vollständige Dateiname „MSBasic Arbeit: Monitor.14".
Wie im UCSD-Pascal kann man mit derselben Syntax unterschiedliche Geräte ansprechen, also nicht nur Dateien auf Disketten. Es gibt folgende vordefinierte Gerätenamen:

**„SCRN:"** (Ausgabedatei) entspricht dem Ausgabefenster.
**„KYBD:"** (Eingabedatei) entspricht der Tastatur.
**„LPT1:"** (Ausgabedatei) entspricht dem Drucker.
**„CLIP:"** (Ein- oder Ausgabedatei) entspricht dem Text-Zwischenspeicher (s. 1.2.2.). Diese Datei kann dazu benutzt werden, um Daten zwischen einem Basic-Programm und einem anderen Programm (z.B. Multiplan von Microsoft) zu übertragen. Dies ist deshalb möglich, weil der Text-Zwischenspeicher nicht nur vom Basic, sondern vom normalen Macintosh-Betriebssystem unterstützt wird. Er wird auch nicht gelöscht, falls ein Programm beendet und ein anderes aufgerufen wird.

**„COM1:"** (Ein- oder Ausgabedatei) entspricht der seriellen Schnittstelle des Macintosh. Hiermit können andere an den Macintosh angeschlossene Peripheriegeräte angesprochen werden, z.B. ein anderer Macintosh oder ein Apple. Ein entsprechenes Programm erscheint in einer der nächsten „Peeker"-Ausgaben.

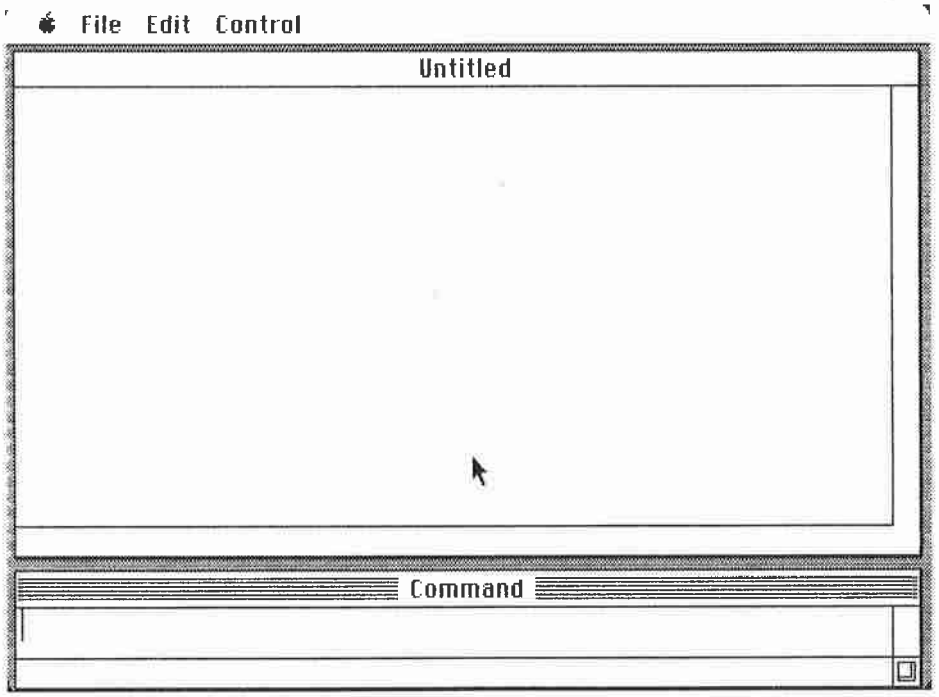


*Abb 1: Schreibtisch am Anfang*

## 2. OPERANDEN

### 2.1. Konstanten

Es gibt beim Microsoft-Basic zwei verschiedene Arten von Konstanten: Stringkonstanten (Zeichenketten) und numerische Konstanten (Zahlen). Bei den numerischen Konstanten kann man noch einmal drei verschiedene Typen unterscheiden.

*2.1.1. Stringkonstanten*

Eine Stringkonstante besteht aus einer Folge von alphanumerischen Zeichen, die von Anführungszeichen (") eingeschlossen sind (wie beim Applesoft-Basic). Strings können eine Länge von bis zu 32767 (!) Zeichen haben.

*2.1.2. Integerkonstanten*

Dies ist die erste Art von numerischen Konstanten. Es handelt sich hier wie beim Applesoft-Basic um ganze Zahlen aus dem Bereich von -32768 bis 32767. In diesen Konstanten kommt kein Dezimalpunkt vor.
Integerkonstanten können auch in hexadezimaler oder in oktaler Schreibweise angegeben werden. Hexadezimale Integerkonstanten beginnen mit den Zeichen „&H", oktale Konstanten mit den Zeichen „&O". Falls eine dieser Konstanten nicht im Bereich -32768 bis 32767 liegt, erscheint die Meldung „Overflow Error".

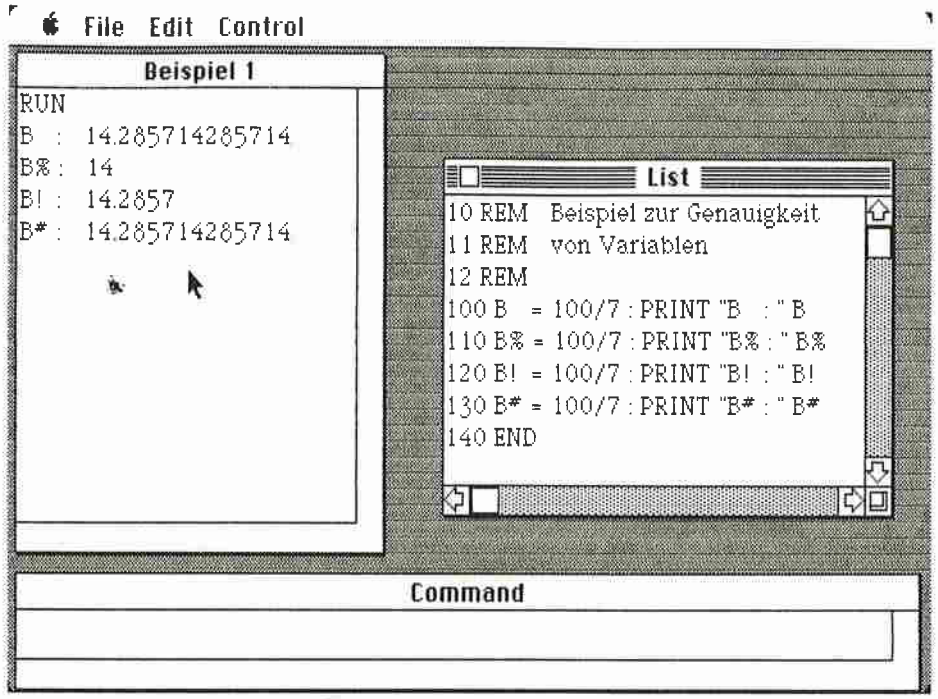
*2.1.3. Festkomma-Konstanten*

Hierbei handelt es sich um reelle (genauer: um rationale) Zahlen, also um Zahlen, die einen Dezimalpunkt beinhalten. Der Zahlenbereich für Festkomma-Konstanten liegt bei 10 ↑ -64 bis 9.9999999999999 · 10 ↑ 62.

*2.1.4. Gleitkomma-Konstanten*

Auch hierbei handelt es sich um rationale Zahlen, die aber in exponentieller Form dargestellt sind. Das bedeutet, sie bestehen aus einer Integer- oder einer Festkomma-Konstante (Mantisse), gefolgt von dem Zeichen „E" und einer ganzen Zahl (Exponent). Statt des Zeichens „E" kann auch das Zeichen „D" stehen (s. 2.1.5.).

*Abb 2: Genauigkeit von Variablen*




Der Zahlenbereich ist wie bei Festkomma-Konstanten 1E-64 bis 9.9999999999999 D62.

*2.1.5. Einfache oder doppelte Genauigkeit*

Numerische Konstanten können entweder mit einfacher oder mit doppelter Genauigkeit gespeichert und ausgegeben werden. Zahlen mit einfacher Genauigkeit werden mit bis zu 6 gültigen Ziffern gespeichert und dargestellt, Zahlen mit doppelter Genauigkeit mit bis zu 14 gültigen Ziffern. Eine numerische Konstante mit einfacher Genauigkeit hat eine der folgenden Eigenschaften:

– 6 oder weniger Ziffern

– exponentielle Form mit dem Zeichen „E"

– nach der Konstante folgt ein „!"

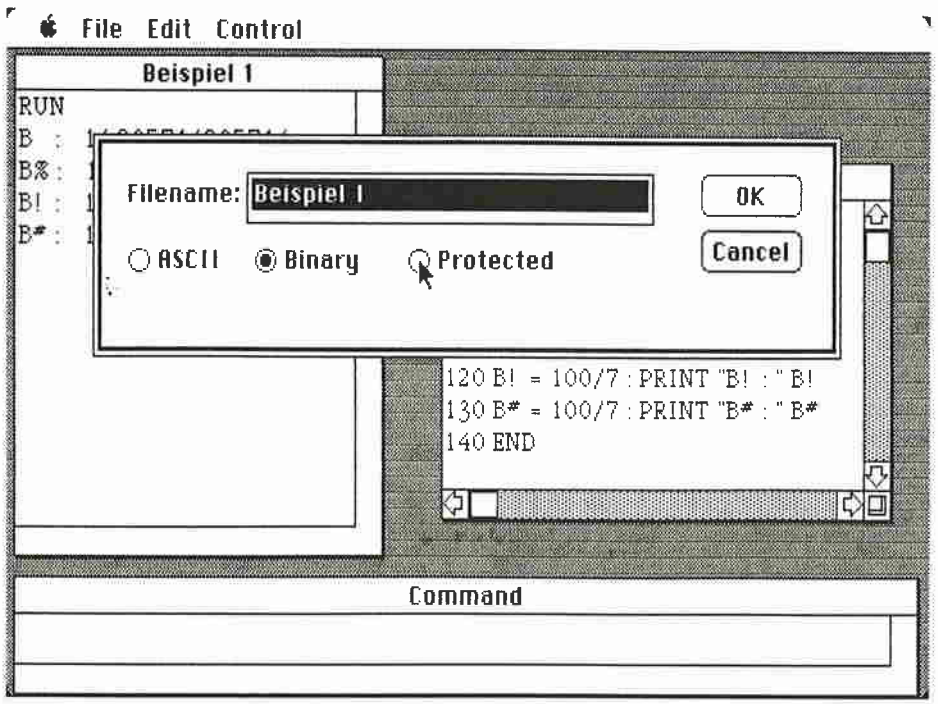



*Abb 3: Beispiel für Dialogfenster*

Beispiele: 1234, 9823.23E-5, 1.2345678!, 3.141 usw.

Doppelte Genauigkeit haben numerische Konstanten mit einer der folgenden Eigenschaften:

– 7 oder mehr Ziffern

– exponentielle Form mit dem Zeichen „D"

– nach der Konstante folgt das Zeichen „#"

Beispiele: 123456789, -4.52D22, 1#, 13.333333 usw.

## 2.2. Variablen

Wie bei den Konstanten gibt es auch zwei verschiedene Arten von Variablen: String- und numerische Variablen. Bei den numerischen Variablen gibt es ebenfalls wie bei den Konstanten unterschiedliche Typen. Jeder Variablen kann nur ein solcher Wert zugewiesen werden, der vom selben Typ ist (vgl. auch 4.3.). Bevor einer Variablen ein Wert zugewiesen wurde, ist ihr Wert "" bei String- und 0 bei numerischen Konstanten.

Beim Microsoft-Basic kann der Name einer Variablen bis zu 40 Zeichen lang sein. Dabei sind alle 40 Zeichen signifikant, d. h. zwei Variablennamen, die sich erst im 40. Zeichen unterscheiden, stehen für zwei verschiedene Variablen (beim Applesoft-Basic: beliebige Zahl von Zeichen, davon jedoch nur 2 signifikant).

Das erste Zeichen muß ein Buchstabe sein; dahinter sind auch Zahlen und der Punkt („.") erlaubt. Kleinbuchstaben werden bei der Eingabe automatisch in Großbuchstaben umgewandelt. Es gibt bestimmte Namen, die nicht als Variablennamen verwendet werden dürfen. Das sind zum einen die Namen von bestimmten ROM-Routinen, die von Basic unterstützt werden, wie z.B. „Line" (wird später erklärt). Zum anderen werden alle Variablen, deren Name mit den Zeichen „FN" beginnt, als Aufruf von benutzerdefinierten Funktionen interpretiert. Auch reservierte Worte wie z.B. „END" können natürlich nicht als Variablennamen verwendet werden. Sie können aber – im Unterschied zum Applesoft-Basic – *innerhalb* der Variablennamen vorkommen. So ist z.B. „ENDE" ein korrekter Variablenname.

*2.2.1. String-Variablen*

Eine String-Variable ist dadurch gekennzeichnet, daß hinter ihrem Namen ein „$"-Zeichen steht. String-Variablen können wie die String-Konstanten bis zu 32767 Zeichen lang sein. Der Wert einer String-Variablen verbraucht im Speicher 5 + n

Bytes, wobei n die Länge des aktuellen Inhalts der Variablen ist.

*2.2.2. Integer-Variablen*

Eine Integer-Variable ist dadurch gekennzeichnet, daß hinter ihrem Namen das Zeichen „%" folgt. Der Wert einer Integer-Variablen muß im Bereich -32768 bis 32767 liegen. Er benötigt 2 Bytes Speicherplatz.

*2.2.3. Variablen mit einfacher Genauigkeit*

Hier steht nach dem Namen der Variablen das Zeichen „!". Der Wert einer solchen Variablen ist eine rationale Zahl aus dem Bereich von 1E-64 bis 9.99999 E 62. Er wird mit einer Genauigkeit von bis zu 6 Ziffern gespeichert. Im Speicher werden dadurch 4 Bytes belegt.

*2.2.4. Variablen mit doppelter Genauigkeit*

Wenn hinter dem Namen einer Variablen kein besonderes typdefinierendes Zeichen steht, so wird vom Microsoft-Basic angenommen, daß es sich um eine Variable mit doppelter Genauigkeit handelt (vgl. aber 2.2.5.). Man kann eine solche Variable aber auch mit dem Zeichen „#" hinter dem Variablennamen kennzeichnen. Der Zahlenbereich dieser Variablen reicht von 1E-64 bis 9.9999999999999 D62. Es werden 8 Bytes im Speicher belegt.

*2.2.5. Änderung des „Default-Typs"*

Man kann mit den Basic-Befehlen „DEFSTR", „DEFINT", „DEFSNG" und „DEFDBL" bestimmen, welchen Typ eine Variable hat, die hinter ihrem Namen *kein* typdefinierendes Zeichen hat. Normalerweise wird in einem solchen Fall angenommen, daß es sich um eine Variable mit doppelter Genauigkeit handelt (s. 2.2.4). Mit den obengenannten Basic-Befehlen kann man dies so abändern, daß Variablen, die mit bestimmten Buchstaben beginnen, von einem bestimmten Typ sind, Variablen mit einem anderen Anfangsbuchstaben aber von einem anderen Typ. Dies wird noch näher bei den Basic-Befehlen in den nächsten Folgen dieser Serie erklärt.
Beispiele zu den Genauigkeiten der einzelnen Variablentypen sind in Abb. 2 zu sehen. In dem Listfenster steht das „Programm", im Ausgabefenster die Ausgabe dieses Programms. Beide Fenster wurden in ihrer Größe verändert, um Listing und Ausgabe nebeneinander und nicht übereinander darstellen zu können.

*2.2.6 Arrays*

Wie beim Applesoft-Basic können mehrere Variablen desselben Typs unter einem Namen zu einem Array zusammengefaßt werden. Der Typ der einzelnen Elemente wird wie oben beschrieben durch den Namen des Arrays festgelegt.
Beim Microsoft-Basic können Arrays bis zu 255 Dimensionen haben. Jede Dimension kann bis zu 32 768 Elemente umfassen. Diese Angaben sind dem Handbuch entnommen. Sie konnten aber nicht nachgeprüft werden, da zum einen eine Eingabezeile nur 255 Zeichen lang sein kann, zum anderen zu wenig Speicher zur Verfügung steht.

## 3. OPERATOREN

Da es zwei verschiedene Arten von Konstanten und Variablen gibt, gibt es auch zwei verschiedene Arten von Operatoren: solche, die Strings als Operanden benötigen, und solche mit numerischen Operatoren.

### 3.1. String-Operatoren

„+": Mit dem Pluszeichen werden wie beim Applesoft-Basic zwei Strings aneinandergehängt (= verkettet).
Außer diesem Operator gibt es für Strings nur noch vergleichende Operatoren. Diese werden im Abschrift 3.3. für numerische Operanden erklärt, haben bei Stringoperanden aber dieselbe Bedeutung. Die Strings sind wie beim Applesoft-Basic nach dem Alphabet geordnet.

### 3.2. Arithmetische Operatoren

Diese Operatoren verknüpfen zwei numerische Operanden zu einem neuen numerischen Wert. Es gibt die folgenden Operatoren:
„↑" – Potenzierung
„*" – Multiplikation
„/" – Division
„\" – ganzzahlige Division
„+" – Addition
„-" – Subtraktion
„**MOD**" – Modulo-Arithmetik

Die Operatoren „\" und „MOD" erfordern, daß beide Operanden bei der Auswertung im Bereich -32768 bis 32767 liegen, ansonsten erfolgt ein „Overflow Error".
Nachzutragen bleibt noch, daß es selbstverständlich auch den Operator „-" (arithmetische Negation) für nur einen Operanden gibt (z. B. „-×").

### 3.3. Vergleiche

Diese Operatoren vergleichen zwei numerische Operanden oder zwei Stringoperanden miteinander. Sie liefern als Ergebnis entweder den numerischen Wert -1 (wahr) oder 0 (falsch). Es gibt die folgenden Vergleiche:
„=" – Gleichheit
„< >" oder „><" – Ungleichheit
„<" – kleiner als
„>" – größer als
„< =" oder „= <" – kleiner oder gleich
„> =" oder „= >" – größer oder gleich

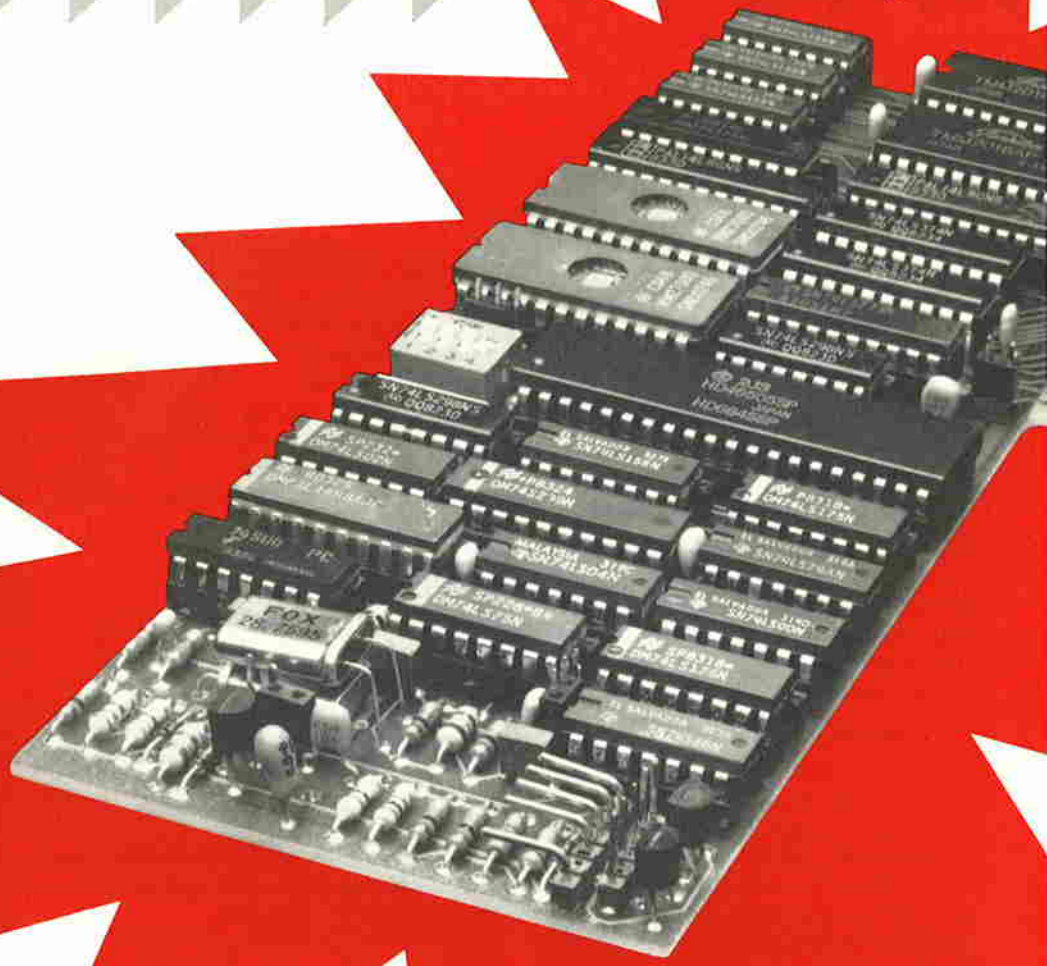
### 3.4. Logische Operatoren

Diese Operatoren verknüpfen zwei numerische Operanden, die im Bereich -32768 bis 32767 liegen müssen. Die beiden Operanden werden im Zweierkomplement als 16-Bit-Zahlen dargestellt. Alle 16 Bits der beiden Operanden werden dann gemäß der gegebenen Operation miteinander verknüpft. Das Ergebnis ist also wieder eine numerische 16-Bit-Zahl aus demselben Bereich. Es gibt die folgenden Operatoren:
„**AND**" – logisches Und
„**OR**" – logisches Oder
„**XOR**" – logisches Exklusiv-Oder
„**IMP**" – logische Implikation
„**EQV**" – logische Äquivalenz

Daneben gibt es noch den Operator „**NOT**", der alle 16 Bits des nachfolgenden Operanden negiert.
Wenn man für den booleschen Wert „wahr" die Zahl -1 und für den booleschen Wert „falsch" die Zahl 0 einsetzt (vgl. 3.3.), dann ist auch das Ergebnis entweder -1 (wahr) oder 0 (falsch). Es ergeben sich dadurch die normalen *booleschen* Operatoren.
Es gilt beispielsweise:
15 AND 14 = 14
4 OR 2 = 6
3 XOR 1 = 2
−X = 1 + NOT X
X = X AND -1
X MOD 2 = X AND 1.

### 3.5. Hierarchie

Falls in einem Ausdruck nicht durch Klammern („(", „)") eine Reihenfolge vorgegeben ist, werden manche Operatoren vor anderen ausgeführt (z.B. Punktrechnung vor Strichrechnung). Es gilt die folgende Hierarchie der einzelnen Operatoren:
1. Potenzierung, 2. arithmetische Negation, 3. Multiplikation und Division, 4. ganzzahlige Division, 5. Modulo-Arithmetik, 6. Addition und Subtraktion, 7. Vergleiche, 8. logische Negation, 9. logisches Und, 10. logisches Oder und Exklusiv-Oder, 11. logische Äquivalenz, 12. logische Implikation



# Gestochen scharf

## Die Videokarte UltraTerm für den Apple II

von Dieter und Jürgen Geiß

**Fast jeder kennt die 80-Zeichen-Karte Videoterm der Firma Videx. Diese Karte ist die am weitesten verbreitete 80-Zeichen-Karte für Apple-II-Computer. Dies ist auch der Grund für die vielen Videoterm-Nachbauten, die es inzwischen auf dem deutschen Markt zu kaufen gibt. Letztes Jahr hat Videx wieder zugeschlagen mit einer ganz neuen 80-Zeichen-Karte, die mehr bietet als jede andere 80-Zeichen-Karte. Es ist die neue UltraTerm für den Apple II, II Plus und IIe. Sie bietet nicht nur eine variable Anzahl von Zeichen pro Zeile, sondern auch eine Schönschrift und vieles andere mehr.**

Als wir uns letztes Jahr in England die Accelerator-Karte bestellten (siehe „Peeker" Heft 1/1984), bekamen wir vom dortigen Händler auch eine Produktinformation über die UltraTerm. Nachdem wir diese gelesen hatten, waren wir begeistert. Daß mit dieser Karte auch einige – wenn auch kleine – Probleme auftauchen würden, erfuhren wir erst, als wir die Karte testeten. Doch dazu später mehr. Ein Freund brachte uns die Karte aus Amerika mit. Nach Umrechnung in deutsche Mark kam diese dann immerhin auf den stolzen Preis von ca. 1100,- DM. Das ist zwar viel für eine 80-Zeichen-Karte, aber nach über einem Jahr Benutzung kann man sagen, daß es sich gelohnt hat. Übrigens kostet diese Karte in den USA inzwischen nur noch etwa $200.00.

Beim Auspacken der UltraTerm-Karte stellt man fest, daß diese trotz der wesentlich höheren Leistung nicht größer als die bisherige Videoterm ist. Der erste Blick fällt auf das größte IC auf der Karte, den CRTC-Prozessor. Es handelt sich hierbei um den HD6845S, den man schon von der bekannten „Videoterm" kennt. Dann findet man noch zwei 2732A EPROMs für den Zeichensatz beziehungsweise für die Firmware der Karte und 4K RAM für die Darstellung von maximal 4096 Zeichen.

Der alte 40/80-Zeichen-Softswitch älterer Versionen der Videoterm existiert nicht mehr. Wer jetzt aber denkt, daß der Bildschirm beim Einschalten des Rechners dunkel bleibt wie bei den 80-Zeichen-Karten ohne Softswitch, der irrt. Zum Zubehör gehört ein spezielles Kabel, das drei Anschlüsse besitzt. Ein Anschluß hat fünf Pins und wird in die UltraTerm gesteckt. Die beiden anderen sind Cinch-Stecker. Einer davon geht in den Apple-Video-Ausgang, und der andere wird mit dem Monitor verbunden. Dadurch ist es möglich, per Programm auf das Apple-Video-Signal (40 Zeichen x 24 Zeilen) umzuschalten, z.B. mit PRINT CHR$ (21), und wieder zurück zum UltraTerm-Video-Signal, z.B. mit PRINT CHR$ (0). Beim Einschalten des Rechners wird natürlich auf das Apple-Video-Signal geschaltet, ebenso bei Reset. Das heißt, auch hier ist kein Umstekken des Video-Kabels mehr nötig.

## Was aber bietet nun die UltraTerm alles?

- 80 x 24 Normalschrift, nicht interlaced
- 96 x 24 Normalschrift, nicht interlaced
- 160 x 24 Normalschrift, nicht interlaced
- 80 x 24 Schönschrift, interlaced
- 80 x 32 Schönschrift, interlaced
- 80 x 48 Normalschrift, interlaced
- 132 x 24 Schönschrift, interlaced
- 128 x 32 Schönschrift, interlaced

(z. B. 128 x 32 = 128 Zeichen mal 32 Zeilen)

- Die alte „Videoterm" wird emuliert
- Zeichensatz-Attribute:
- Normal/Schönschrift
- Normal/Inverse Video
- Helle/Dunkle Zeichen
- Stabiles flimmerfreies Bild
- Block-Graphik
- Sehr gut lesbarer 7 x 9 Punktmatrix Zeichensatz
- Benutzung mit Apple IIe und 64K-Erweiterung möglich

## Inbetriebnahme der UltraTerm

Um die UltraTerm in Betrieb zu nehmen, steckt man das Video-Connect-Kabel mit den fünf Pins in die UltraTerm, den einen Cinch-Stecker in den Video-Ausgang des Apple, und den anderen verbindet man mit dem Monitor. Die Karte selbst installiert man am besten im Slot 3, damit man auch in Pascal damit arbeiten kann. Beim Apple IIe könnte dieser Slot aber nicht zugänglich sein, wenn man die 64K-Erweiterung (Extended 80 Column Card) hat. Auch daran haben die Techniker von Videx gedacht. Es gibt eine Brücke auf der UltraTerm, die man umstecken kann. Damit ist es möglich, die Karte in einem beliebigen Slot zu installieren, wobei die Hardware aber so getäuscht wird, daß sie annimmt, die Karte befinde sich dennoch in Slot 3, d.h. sie kann zum Beispiel mit PR#3 in Basic initialisiert werden. Damit kann die UltraTerm auch auf jeden Fall in Pascal richtig arbeiten.

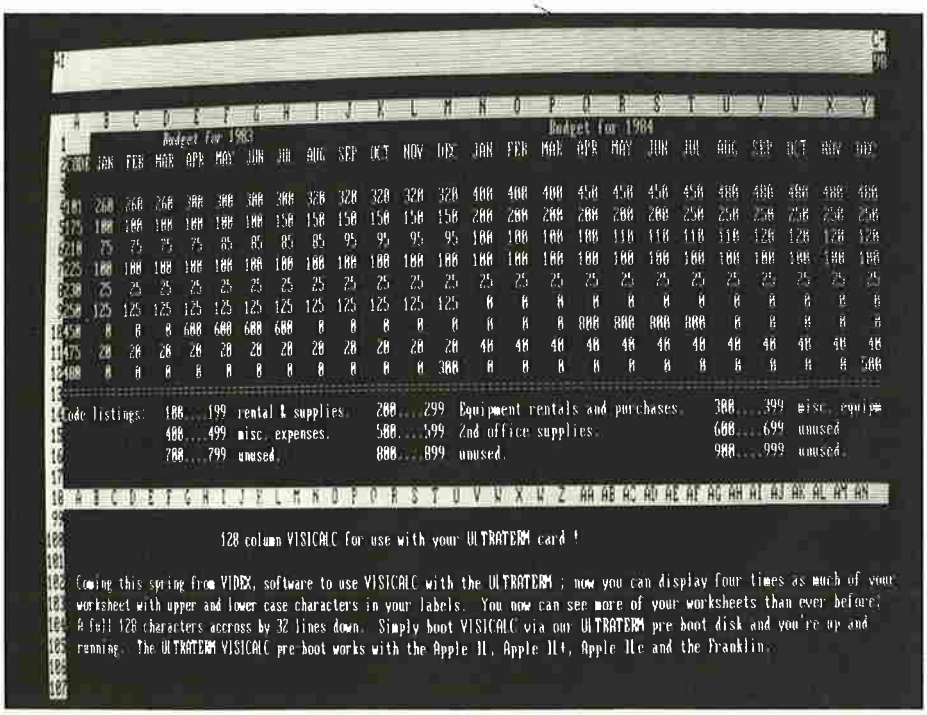
Als wir die UltraTerm eingebaut hatten, wollten wir zuerst einen Überblick über die verschiedenen Schriftarten und -möglichkeiten haben. Zu diesem Zweck legten wir die beigefügte Demodiskette ein, die auf sehr eindrucksvolle Art die Möglichkeiten der UltraTerm aufzeigt. Wir waren allerdings noch nicht ganz zufrieden. Der Grund dafür war unser Monitor, der nicht geeignet war, einige der Möglichkeiten der UltraTerm auszunutzen. Das lag am nicht ausreichend lange nachglühenden Phosphor, den die UltraTerm im Interlace-Modus braucht, damit die Schrift nicht flimmert. Im Interlace-Modus arbeitet die UltraTerm nämlich wie folgt: Der Bildschirm wird hier nicht wie sonst jede 1/60 Sekunde, sondern nur noch jede 1/30 Sekunde erneuert. Dadurch können auf dem Bildschirm doppelt so viele Zeilen mit halbem Zeilenabstand dargestellt werden als sonst. Man kann sich das etwa so vorstellen, als ob in der ersten 1/30 Sekunde die geraden Zeilen und in der zweiten 1/30 Sekunde die ungeraden Zeilen geschrieben werden. Während der zweiten 1/30 Sekunde glühen also die Zeilen der ersten 1/30 Sekunde aus. Dadurch ist es notwendig, einen Monitor mit lange nachglühendem Phosphor zu benutzen.

Im Handbuch werden einige Monitore beschrieben, die diesen Phosphor haben, und einer davon ist der Apple Monitor III, der von Videx sehr empfohlen wird. Da wir diesen Monitor nicht hatten, nutzte uns die Karte zunächst wenig. Also mußte auch noch einen Monitor III gekauft werden. Als wir jedoch den neuen Monitor anschlossen, bot sich ein erschreckendes Bild: Die Buchstaben waren sehr unscharf und schlechter zu lesen als vorher. Was war

passiert? Nach genauen Studien des Handbuches stießen wir auf die richtige Stelle. Der Jumper J4 sorgt dafür, daß das Videsignal eine rechteckige oder dreieckige Wellenform hat. Der Monitor III kann aber mit den dreieckigen Signalen mehr anfangen, und als wir den Jumper umsteckten, war die Welt wieder in Ordnung. Mit dem neuen Monitor konnten wir nunmehr auch die Interlace-Modi ausnutzen. Im Interlace-Modus sind die Buchstaben halb so hoch, und die einzelnen Punkte, aus der die Matrix der Buchstaben besteht, sind dann nicht mehr zu erkennen. Es ergibt sich ein homogenes Schriftbild ähnlich dem einer Schreibmaschine. Statt der 7 x 9 – Matrix wird eine 9 x 16 – Matrix benutzt, in der die Buchstaben feiner definiert wurden. Aber auch die normale 7 x 9 – Matrix unterscheidet sich von der alten Videoterm. Die Buchstaben erscheinen nicht so dünn, sondern sind doppelt so kräftig in der Breite, d.h. statt einem Punkt sind jeweils zwei Punkte nebeneinander definiert. Da auch inverse Schrift implementiert ist, fehlt die Umschaltung auf den deutschen Zeichensatz.
Wir haben uns geholfen, indem wir ein zweites EPROM auf das erste gesteckt und einen Umschalter an der Karte angebracht haben. Als Pascal-Programmierer brauchen wir natürlich eher die eckigen oder geschweiften Klammern, aber darüber hinaus sind auch Umlaute für Textverarbeitung erforderlich. Wer ausschließlich in Deutsch arbeiten will, kann das vorhandene Eprom gleich gegen das deutsche austauschen. Mit einem Eprommer kann man das 2732A programmieren. Der von uns erstellte deutsche Zeichensatz wird auf der Peeker-Sammeldiskette zu finden sein.

## Arbeiten mit der UltraTerm unter Applesoft-Basic

Das Arbeiten unter DOS 3.3 oder ProDOS ist sehr angenehm. Basic-Zeilen können – wie die Erfahrung lehrt – sehr lange werden. Beim Auflisten eines Programms auf einem 40-Zeichen-Bildschirm besteht nur noch eine sehr schlechte Übersicht. Mit dem Modus 128 Zeichen mit 32 Zeilen paßt aber sehr oft eine ganze Basic-Zeile in eine Bildschirm-Zeile. Damit ist auch das Editieren durch Überfahren mit dem Cursor sehr leicht zu bewerkstelligen. Verschiedene Applesoft-Befehle funktionieren jetzt richtig.
Bei der alten Videoterm konnte man durch HOME lediglich den Cursor in die linke obere Ecke positionieren. Der Bildschirm wurde nicht gelöscht, wenn man sich im 80-Zeichen-Modus befand. Bei der UltraTerm wird auch der Bildschirm gelöscht. Der Befehl INVERSE ergibt nun tatsächlich inverse Buchstaben; mit NORMAL gelangt man wieder zur Normalschrift. Außerdem kann man einzelne Buchstaben heller aufleuchten lassen und so gewisse Stellen in einem Text besonders hervorheben. Mit Ctrl-A (ASCII-Code 1) kann man auf Kleinbuchstaben umschalten. Empfängt die UltraTerm auch nur einen einzigen Kleinbuchstaben über die RDKEY-Tastaturabfrage, d.h. ist es der Tastatur des Apple II Plus möglich, Kleinbuchstaben zu erzeugen, so wird diese spezielle Ctrl-A-Funktion wieder aufgehoben und Ctrl-A ergibt den normalen ASCII-Code 1.
Beim Apple IIe oder beim Apple II Plus mit Tastaturumrüstsatz sind alle Zeichen von der Tastatur aus eingebbar. Alle Cursorsteuertasten ESC A, B, C, D sowie ESC I, J, K, M sind implementiert und funktionieren auch mit den entsprechenden Kleinbuchstaben. Dasselbe gilt im übrigen für alle Basic- oder DOS-Befehle. Alles, was nicht in Hochkommata eingeschlossen ist, wird in Großbuchstaben umgewandelt, so daß man überhaupt nicht auf SHIFT-LOCK umschalten muß, wenn man in Basic programmiert.



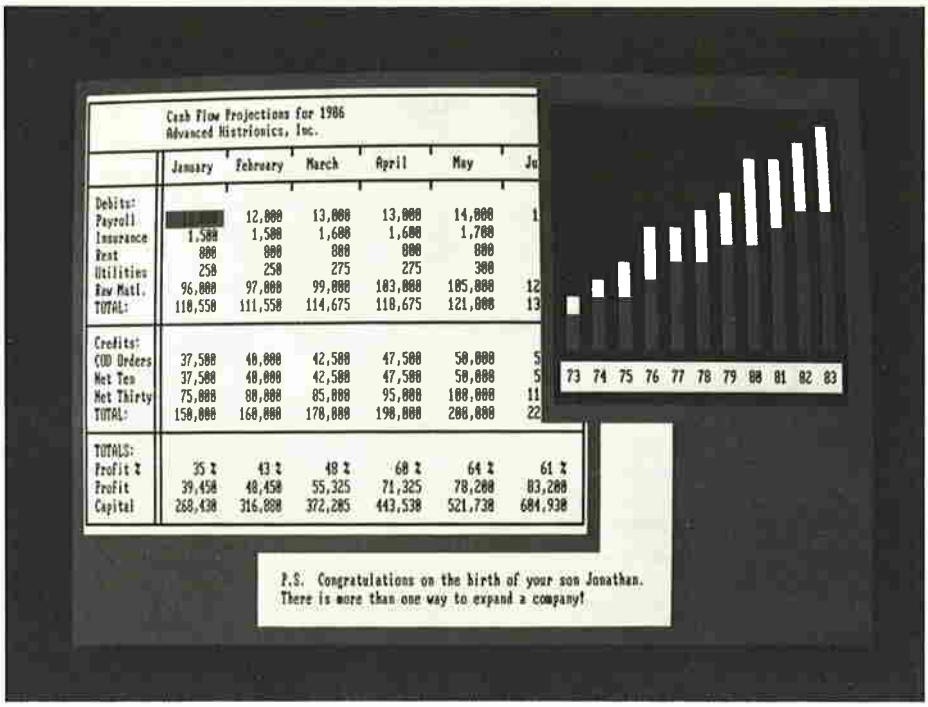

Es wurde großer Wert auf Kompatibilität mit dem 40-Zeichen-Bildschirm in Basic gelegt. Im TEXT-Modus ist dies auch bis auf den FLASH-Befehl gelungen. Schaltet man mittels HGR auf Grafik um, ist man jedoch immer noch auf der Text-Seite. Dies liegt an dem fehlenden Softswitch, der allerdings softwaremäßig simulierbar ist. Mit PRINT CHR$ (22) „0" wird auf den Standard-Bildschirm und damit auf die Grafik-Seite umgeschaltet; danach kommt man auf die Text-Seite mit PRINT CHR$ (0).

## Arbeiten mit der UltraTerm unter UCSD-Pascal

Da die Ein/Ausgabe in Pascal hervorragend implementiert wurde, gibt es beim Benutzen der neuen Karte keine Schwierigkeiten im 80 x 24 Modus. Sogar der Type-ahead-Buffer – Zeichen können vorausgetippt werden – funktioniert sofort, was bei der alten Videoterm nicht ohne Patch möglich war. Dies liegt daran, daß das Protokoll für Interface Karten endlich eingehalten wurde und damit die Karte vom Typ „Firmware" von Pascal erkannt wird.
Aber es tauchten einige Probleme mit den neuen Bildschirmformaten auf, insbesondere mit den Formaten mit mehr als 80 Zeichen pro Zeile. Wie Pascal-Programmierer wissen, kann das Pascal-System nahezu an jede Tastatur und jeden Bildschirm angeschlossen werden. Die APPLE3-Diskette enthält das dafür notwendige Programm SETUP. Möchte man beispielsweise einen 128 x 32 Zeichen Bildschirm benutzen, so kann man dies im SETUP-Programm angeben. Man verändert einfach Screen Width beziehungsweise Screen Height. Anschließend müssen

dann noch die entsprechenden Steuerzeichen an die UltraTerm ausgegeben werden. Dies kann gut vom F(iler aus mittels T(ransfer #1,#1 durchgeführt werden.
Da nunmehr alle notwendigen Patches erledigt sind, kann man in den Editor gehen und Texte mit bis zu 128 x 32 Zeichen auf einer Bildschirmseite editieren. Wer dies mit dem SYSTEM.EDITOR der Pascal Version 1.1 oder 1.2 versucht, erlebt aber eine herbe Enttäuschung. Wenn man nämlich in eine Zeile, die schon 80 oder mehr Zeichen lang ist, noch ein paar Zeichen einfügen will, so bootet das System neu oder meldet sich wenigstens mit STACK OVERFLOW, PRESS RESET. Warum funktioniert das nicht?

Die Leute von der Universität von Kalifornien in San Diego oder kurz UCSD hatten bei der Entwicklung des Editors vor ca. 6 Jahren natürlich nicht daran gedacht, daß ein Bildschirm einmal mit mehr als 80 Zeichen/Zeile betrieben werden könnte und testeten dies nie aus. Wenigstens ist nicht die UltraTerm daran schuld. Was aber tun? Wer im Besitz des Modula-Systems für den Apple ist, der kann aufatmen. Auf diesen Systemdisketten befindet sich der ASE (Advanced System Editor), der mit der UltraTerm einwandfrei arbeitet. Wer einmal mit dem ASE editiert hat, der weiß, daß der alte Editor ohnehin nicht der beste war.
Ein weiterer Nachteil ergibt sich, wenn man in Pascal mit der Turtle Graphic arbeiten möchte. Läßt man z.B. das Programm GRAFDEMO von der APPLE3-Diskette laufen, so schaltet der Bildschirm wegen des fehlendes Softswitches nicht auf den Graphikschirm um und man sieht gar nichts. Programmiert man selbst, so kann man, nachdem man mit dem Befehl Grafmode auf Grafik umgeschaltet hat, zusätzlich noch ein WRITE CHR (21) an die UltraTerm ausgeben. Dies aktiviert dann den Grafik-Bildschirm.
Will man aber beispielsweise Zeichen von der Tastatur einlesen, während man sich in der Grafik befindet, so schaltet die UltraTerm leider immer wieder auf Text um. Dies liegt daran, daß ein Initialisierungsunterprogramm der UltraTerm sowohl bei READ- als auch bei WRITE-Anweisungen ausgeführt wird, das immer wieder auf Text schaltet. Vielleicht wird dies bei einer zukünftigen UltraTerm-Firmware behoben. Wer aber schon die UltraTerm besitzt und trotzdem mit Grafikanwenderprogrammen zu tun hat, kann dieses Manko auch anders beheben. Inzwischen existiert eine neue Turtle Graphics Unit, die von uns speziell für die UltraTerm modifiziert wurde und eine Reihe weiterer mächtiger Features aufweist. Diese neue Turtle Graphics wird in einem der nächsten Hefte von „Peeker" vorgestellt. Damit laufen dann auch fremde Grafikprogramme.




## Arbeiten mit der UltraTerm und einem Assembler

Den großen Überblick bekommt man durch die UltraTerm, wenn man mit einem Assembler wie etwa Big Mac (Merlin) arbeitet. Eine 80-Zeichen-Karte wird dort ohnehin unterstützt und mehr als 80 Zeichen sind bei einem Assembler nicht erforderlich. Aber man hat ja den Modus 80 Zeichen auf 48 Zeilen; dies sind doppelt soviele Zeilen als üblich.

## Die UltraTerm und einige spezielle Programme

### Visicalc

Von der Firma Videx wird eine Visicalc-Preboot-Diskette angeboten, die mehrere Modi zuläßt. Benutzt man den 128 x 32 Modus, so passen mehr als doppelt so viele Felder auf eine Bildschirmseite. Für die Preboot-Diskette braucht man eine 16-Sector-Version von Visicalc mit einer Versionsnummer von 193, 202 oder 208.

### Wordstar

Um die UltraTerm mit Wordstar zu benutzen, muß man zuerst das Programm INSTALL laufen lassen, das mit Wordstar mitgeliefert wird. Da Wordstar aber grundsätzlich nur mit höchstens 80 Zeichen arbeitet, sollte man den 80 x 48 Modus benutzen. Dies bringt immerhin 3840 Zeichen auf den Bildschirm.

### Applewriter

Auch für den Applewriter wird eine Preboot-Diskette angeboten. Zur Druckzeit des UltraTerm-Handbuchs wurde allerdings noch keine genaue Spezifikation für diese Diskette angegeben.

Das Handbuch ist sehr umfangreich und enthält Information über alles, was man über die UltraTerm wissen möchte, wie z.B. die Programmierung der CRTC-Register u.a. Die Firmware ist als sauber dokumentiertes Assemblerlisting im Anhang enthalten, und ein Bildschirmtreiber wird als Beispiel vorgestellt. Ein Schaltbild gehört ebenso zur Dokumentation wie die Beschreibung der Arbeitsweise der Karte. Das Handbuch ist in Englisch.
Abschließend ist zu sagen, daß die UltraTerm die einzige Karte mit mehreren wählbaren Video-Modi ist. Man kann mit ihr wesentlich mehr Zeichen auf dem Bildschirm darstellen als mit anderen Karten. Außerdem ist bei ihr das Schriftbild ungleich schärfer und besser als etwa bei der 80-Zeichenkarte des Apple IIe.

# Prometric

## Ein Apple-Kompatibler mit 65C02C

Das in Produktion befindliche Gerät Prometric der Firma Dreieich Rechner Versand DRV, Dr. Böhmer in 6072 Dreieichenhain, Am Kellersbusch 4, ist u.U. der erste Apple-Kompatible, bei dem der 65C02C mit 3,58 MHz im Motherboard integriert ist. Der Rechner soll im Januar 1985 auf den Markt kommen.

Das Motherboard B2/B3 ist eine komplette Neuentwicklung. Ziel dieser Entwicklung war es, ein hochintegriertes Board zu haben, auf dem bestehende 6502- und Z80-Software mit hoher Geschwindigkeit verarbeitet werden kann. Um auf die bestehende umfangreiche Software-Bibliothek des Apple zugreifen zu können, entschied sich DRV für dieses Format. Ein integrierter Z80B (6 MHz) ermöglicht es, CP/M 2.2 Software zu verarbeiten. Gegen Aufpreis ist eine Z80H-Karte (7.4 MHz) erhältlich, die den Betrieb von CP/M 3.0 zuläßt.

Es wurde ein neues DOS entwickelt, das ca. achtmal schneller als DOS 3.3 ist. Auch zum neuen ProDOS soll dieses DOS voll kompatibel sein.

Entsprechende Patches für CP/M 2.2, CP/M 3.0 und USCD-Pascal sind vorhanden, daher stehen in Verbindung mit entsprechenden Laufwerken pro Diskette 640K formatiert zur Verfügung.

In das Motherboard wurden 3 Prozessoren integriert:
6502 mit 1.00 MHz
65C02C mit 3.58 MHz
Z80B mit 6.00 MHz

Die Erhöhung der Taktfrequenz von 65C02C und Z80B bringt eine Beschleunigung der internen Verarbeitung um den Faktor 3.58 bzw. 6 für DOS-, CP/M- und Pascal-Programme.

Es besteht die Möglichkeit, Applesoft in einem EPROM abzulegen oder von Diskette zu booten. Ausgeliefert wird der Computer mit deutschem BASIC und dem speziellen DOS. Das deutsche DOS ist Applesoft-kompatibel.

Weiterhin sind folgende Hardware-Komponenten in das Motherboard integriert:

*Slot 1:* V24-Schnittstelle seriell und parallele Graphik-Centronics-Schnittstelle, anzuwählen über DIL-Schalter.

*Slot 2:* Frei belegbar oder, über DIL-Schalter, mit integrierter Pseudo-Floppy 256K.

*Slot 3:* Integrierte 80-Zeichen-Karte mit 4 Zeichensätzen, RGB-Farbausgang und Softswitch.

*Slot 4:* Integrierter 65C02 mit zusätzlichen 64K (3.58 MHz) oder, über DIL-Schalter wählbar, integrierter Z80B mit zusätzlich 64K (6 MHz).

*Slot 5, 6, 7:* Stehen zur freien Verfügung, Slot 6 kann mit dem von DRV lieferbaren Controller für 2 Laufwerke 2 · 80 Tracks bestückt werden. Der Betrieb von Apple-Laufwerken an diesem Controller ist möglich, ebenso Mischbetrieb.

*Slot 7:* Kann mit einer PAL-Karte belegt werden.

Auf dem B3-Motherboard ist die Pseudo-Floppy mit 256K voll bestückt, auf dem B2-Motherboard befinden sich die unbestückten Sockel.

Die Maße des Motherboards sind 425 · 300 mm. Es wird als Einzelboard oder eingebaut in ein Tischgehäuse bzw. Portable verkauft.

Das Portable ist 465 B · 195 H · 410 T groß. Der Monitor hat eine Größe von 9". Als Tastatur wird die Preh-Commander mit deutschen Umlauten verwendet.

Im nächsten oder übernächsten Heft von „Peeker" wird ein ausführlicher Test- und Erfahrungsbericht erscheinen.

# TOMBSTONE-MICRO

Th. Tank & G. Körber

| | |
|---|---|
| **MESON II,** 48 K | 1100,– DM |
| **MESON II,** 64 K (16 K + Z 80-Card) | 1250,– DM |
| beide Rechner mit 10er-Block, 1 Jahr Garantie | |
| **Z 80-Card** | 125,– DM |
| **16 K-Card** | 125,– DM |
| **Floppy-Disk** 5¼" | 480,– DM |
| **Controller-Card** | 150,– DM |
| **PAL-Card** + Modulator | 170,– DM |
| **MESON II,** Leerplatine | 85,– DM |
| Tastaturen und Gehäuse auf Anfrage | |

| | |
|---|---|
| **Joyport,** zum Anschluß von zwei ATARI-Joysticks | 40,– DM |
| **ADD2 B,** (VIA 6522) mit RAM + Backup | 170,– DM |
| **ADD2 L,** Leerplatine | 55,– DM |
| **ADD4 B,** (PIA 6821) mit RAM + Backup | 150,– DM |
| **ADD4 L,** Leerplatine | 45,– DM |
| PIA-Card mit Wrap-Feld ausgerüstet | |

**Entwicklungen auf Anfrage**

Gardeschützenweg 72, 1000 Berlin 45
☏ 030/8 33 13 03 (SHOP), Q 7 46 57 28 (BÜRO)

## Charakter-Analyse

### Lernen Sie Ihre Mitmenschen kennen

Mit diesem absolut neuen Programm sind Sie in der Lage, Bekannte, Verwandte, Freunde und Feinde ab sofort richtig beurteilen zu können!
Das Programm bietet Ihnen:
Acht Bildschirmseiten Auswertung der charakteristischen Eigenschaften sowie Farbgrafik!!
Detaillierte Programmerläuterung nach dem Booten.
Eine kundenspezifische Druckversion ist auch erhältlich, bitte Druckertyp angeben!
Ab Januar '85 bieten wir eine Version für den Sprachsynthesizer der Fa. ECKL-ELECTRONIC an!

| | |
|---|---|
| Bildschirmversion für APPLE | DM 59,– |
| Druckerversion für APPLE | DM 98,– |
| Synthesizerversion für APPLE | DM 115,– |

Alle Preise incl. Porto und Verpackung, bar oder Scheck, Nachnahmesendungen zuzüglich DM 9,–

**Snoopy-Soft, T. Jenczyk, Postf. 65, 3250 Hameln 1**

Telefon (02 41) 3 49 62
Noppiusstraße 19, 5100 Aachen

**RÖCKRATH MICROCOMPUTER**

**Das deutsche Apple-II-ROM-Listing ist da!**

- Einleitung zum prinzipiellen Ablauf des Applesoftinterpreters:
  - Aufbau und Verarbeitung der/des
    - Programmtextes – Variablentabelle – Stringspace
    - Fließkommaformate – Basicstacks (GDSUB, FOR-NEXT, ...)
  - Beschreibung der wichtigsten Unterprogramme, z. B. Variablensuche, Garbage collection, Ausdrucksauswertung, CHRGET, ...
- Vollständig disassemblierte und sehr ausführlich deutsch kommentierte Auflistung des Applesoft-BASIC-Interpreters
- Übersichtliche Auflistung aller vom Interpreter benutzten RAM-Zellen mit allen Verwendungszwecken
- Über 150 ausführlich dokumentierte Unterprogramme:
  - Funktion – Ein-/Ausgabeparameter

Auch für Apple-IIe und c und Kompatible!

**Außerdem folgende ROM-Listings vorrätig:**

| | |
|---|---|
| TRS-80 Model I, Genie I+II | 69,50 DM |
| TRS-80 Model III | 79,– DM |
| Colour-Genie | 59,– DM |

Preis inkl. MwSt./Händleranfragen erwünscht!

M. Buck: Apple-II-ROM-Listing (116 S. DIN A4, kartoniert, 59 DM, Best.-Nr.: 06-010-8)

Der Btx.-Fachmann in Ihrer Nähe — Computer Ring

## Floppy-Speicher

applekompatibel NF555
**140 - 250 KB**
geräuschlos durch Direkt-Antrieb, autom. Geschwindigkeitsregelung.
**DM 595,-**

NEWTECH-Qualität

Computer-Studio,
Ringstraße 70,
2300 Kiel 1,
Telefon 04 31/67 66

**Für eilige Anzeigen**
**Telefon 0 62 21 / 48 92 18**
**Telex 461 727**

# S.A.M. The Software Automatic Mouth

## oder wie man den Apple zum Sprechen bringt

**von Ralf Augspurger**

The Software Automatic Mouth ist eine von der Firma Don't Ask in Kalifornien für den Apple II/II+ entwickelte Sprachausgabe. Das Vokabular von S.A.M. ist laut Hersteller fast unbegrenzt.

S.A.M. besteht aus einem Softwarepaket und einem 8-Bit-D/A-Wandler, der sich auf einer kleinen Platine befindet, die in jedem der 7 Slots des Apples installierbar ist. An der Platine ist zusätzlich ein Kabel befestigt, mit dem die gesamte Ausgabe des Computers über den Wandler geleitet werden kann. Dies hat aber für Musikfans den Nachteil, daß die hohen Töne unterdrückt werden.

Nach der Installation der Karte kann die Diskette im DOS 3.3 Format mit der Software gebootet werden. Damit wird S.A.M. ab der Adresse 29024 (hexadezimal $7160) in den Speicher geladen.
Nach dem Booten erwartet S.A.M. die Karte in Slot 4. Wurde sie in einen anderen gesteckt, so muß mit POKE 38140, Slot die Ausgabeadresse berichtigt werden.
Nun kann der Computer sprechen, jedoch nur wenn jede Eingabe in einer Phonetiksprache erfolgt. Die Eingabe muß in den String SA$ geschrieben und danach mit CALL 38128 (S.A.M.-Programm) aufgerufen werden.

Beispiel:

SA$ = "AY4 AEM AH KUMPYUW3TER"
CALL 38128

Dies würde den Computer veranlassen, „I am a computer" zu sprechen. Die Buchstabenkombinationen des Phonetikalphabets müssen nicht mit der des normalen Alphabets übereinstimmen. Siehe dazu **Tabelle 1**.
Die Ziffern nach einigen Phonemen bewirken eine besondere Aussprache des vorhergehenden Lauts:

1 = sehr gefühlvolle Betonung
2 = sehr nachdrückliche Betonung
3 = ziemlich starke Betonung
4 = normale Betonung
5 = hohe Betonung
6 = keine Betonung
7 = tiefe Betonung
8 = sehr tiefe Betonung

Mit dieser Phonetiksprache kann man S.A.M. z. B. deutsch (mit englischem Akzent) sprechen lassen.
Wem das Auswendiglernen der Buchstabenkombinationen zu umständlich ist, für den gibt es auf der Diskette ein Programm mit dem Namen RECITER, zu deutsch „Vortragskünstler". Dieses Programm erlaubt es, daß der Benutzer ganz normales Englisch in den SA$ String schreiben kann. Die Eintrittsadresse für RECITER ist 38131.
Der Nachteil dieses Programms liegt klar auf der Hand: Es wird ein sehr großer Teil des Hauptspeichers belegt. Davon ist auch die 2. Seite der hochauflösenden Graphik betroffen. RECITER wird ab der Adresse 22688 (hexadezimal $58A0) im Speicher abgelegt. Ebenfalls muß auch S.A.M. sich im Speicher befinden.

Nun noch einen kurzen Blick auf die Beispielprogramme, die sich alle auf der Diskette befinden. Bei dem Spiel GUESSNUM wird eine Zahl zwischen 10 und 100 gesucht. Nach Eingabe einer Zahl antwortet S.A.M. mit „Too low", „Too high" oder vielleicht mit „That's right".
Zwei Demoprogramme sind auch vorhanden, welche genau den Sprachumfang, die Aussprache und sonstige Möglichkeiten von S.A.M. demonstrieren. Ein anderes Programm erlaubt das Austesten verschiedener Eingaben, die in Phonetik oder Englisch erfolgen können. Mit Control-N kann die Geschwindigkeit oder die Stimmlage von S.A.M. gesteuert werden. Diese zwei nützlichen Funktionen können auch ohne dieses Programm mit POKE 38141, Stimmlage und POKE 38142, Geschwindigkeit geändert werden. Zum Beispiel veranlasst
POKE 38141, 30
POKE 38142, 30
S.A.M. sehr hoch und schnell zu sprechen. Die Normalwerte sind:
POKE 38141, 72
POKE 38142, 64.

Das gesamte Paket mit Diskette, D/A-Wandler und einer 38seitigen Beschreibung kostet bei der Firma Don't Ask Computer Software, 2265 Westwood Blvd. Suite B-150, Los Angeles, CA 90064 den stolzen Preis von 469.-- DM. In Deutschland ist S.A.M. u.a. über Pandasoft, Berlin, erhältlich.

| VOWELS | |
|---|---|
| **IY** | feet |
| **IH** | pin |
| **EH** | beg |
| **AE** | Sam |
| **AA** | pot |
| **AH** | budget |
| **AO** | talk |
| **OH** | cone |
| **UH** | book |
| **UX** | loot |
| **ER** | bird |
| **AX** | gallon |
| **IX** | digit |
| **DIPHTHONGS** | |
| **EY** | made |
| **AY** | high |
| **OY** | boy |
| **AW** | how |
| **OW** | slow |
| **UW** | crew |

**Tabelle 1**

# MICROMINT

## VOLLTREFFER MICROMINT

**LASAR 16** – IBM Comp. 128 K, Contr. DRIVER 320 K, Netzteil 15 A **3.990,-**

**LASAR ZE** – Apple comp. 64 K + 12 K ROM + 6502 + Z 80 A **1.432,-**

**Außerdem volles Rückgaberecht innerhalb 10 Tagen ohne Begründung.**

| | Apple | IBM |
|---|---|---|
| ● Mehrzweckgehäuse lt. Abb. | 209,– | 209,– |
| ● Schaltnetzteile Apple 5 A/IBM 15 A | 115,– | 350,– |
| ● Profitastatur dtsch. Boss I | 398,– | 398,– |
| ● Interface | ab 122,– | 400,– |
| ● Alle Komponente lieferbar | lt. Preisliste | |

**Prompte Belieferung von 1000 m² Lagerfläche. Kostenlose Tiefstpreishändlerliste noch heute schriftlich anfordern – großes Angebot an IBM-Comp.**

**Generalimporteur MICROMINT STREIL,** Hochdahler Straße 151, 4006 Erkrath 2, Telex 8589305 mcm

☎ **02104/33024**

## Das Besondere

- Akustikkoppler EPSON, AK (FTZ)
- Telecommunications-Software
- Telefon/Modem-Kombination (Export)
- Rufnummernspeicher-Set (Export)
- Controller und Laufwerke bis 800 kB
- 8″-Technik **mit Funktionsgarantie**
- Aktuelle Literatur

## Beate Vollrath Computer

Kirchstraße 17 + 28 · 4650 Gelsenkirchen
Tel. 02 09-20 92 91 · Mailbox 02 09-27 16 66

## CE-50 Typenrad mit Apple II/IIe Interface.

Für Ausgabe und Eingabe!

**DM 1298,-** incl. MwSt. ab Werk

brother
QUALITÄT AUS ERSTER HAND.

Für DOS, Apple-Pascal, CP/M und Textprogramme dieser Betriebssysteme.
Für alle CE-u. EM-Typen lieferbar.

**interkom**
electronic
**Kock & Mreches GmbH**
**3004 Isernhagen 4**
**Tel. 05139-87393**

## APPLE-USER!

Werden Sie Mitglied im größten Computerclub Europas.
Über 5000 Mitglieder,
55 Regionalgruppen,
15 Arbeitsgemeinschaften,
6 MByte Vereinssoftware,
monatliche Zeitschrift.

Info durch:

**Apple User Group Europe e.V.**

Postfach 11 01 69,
4200 Oberhausen 11

Übrigens,
eine Jahresmitgliedschaft ist ein schönes Weihnachtsgeschenk.

## Akustikkoppler AK 300

300 baud
vollduplex
Originate und Answer
**Netz- und Batteriebetrieb**
variable Gummimuffen für alle gängigen Hörer
**FTZ-Nr. 18.13.1897.00**
**gebühren- und anmeldefrei**

**erhältlich in Computerfachgeschäften und in den Computershops der Warenhäuser und Audio- und Videofachgeschäfte**

Vertrieb:

**SOFTWARE EXPRESS**

Hugo-Viehoff-Str. 84, 4000 Düsseldorf 30, Tel. 02 11/42 43 23

## TELETERM 2+/e

**Die Software für die Datenfernübertragung mit dem Apple IIplus/e und dem Akustik-Koppler. Ohne zusätzliche serielle Karte!!! Die Übertragung erfolgt über den Game Port.**

### Leistungsmerkmale:

- komplette Menuesteuerung
- Empfangen und Senden von Telesoftware
- automatisches Abspeichern der ausgetauschten Daten
- V.24-Interface mit Kabel zum Akustik-Koppler
- ausführliche Bedienungsanleitung mit aktuellen Telefonnummern der öffentlich zugänglichen Datenbanken.

**TELETERM.** Auch Ihr Computer will nicht allein sein...

**TELETERM** erhalten Sie in Computer-Fachgeschäften, in den Computer-Abteilungen der Warenhäuser sowie in Audio- und Video-Fachgeschäften.

**SOFTWARE EXPRESS**

Hugo-Viehoff-Straße 84, 4000 Düsseldorf 30
Telefon 02 11/42 43 23

# apple® & CP/M®

**SOFTWARE HARDWARE LITERATUR**

SpeeDemon von M·c·T

DM 1368.-

Machen Sie Ihren APPLE II+/e bis 3,5 x schneller !!
mit CPU 65C02C - 64 K RAM - belegt einen Slot -
Nur ein Beispiel unserer Leistungsfähigkeit! Verlangen Sie Qualität - wir bieten sie - Ihr Import-Spezialist!

**Karl-Heinz Weiß**
Am Wiesenhof 17 · 2940 Wilhelmshaven
☎ 04421/83 179

Katalog gegen eine Schutzgebühr von DM 3.- in Bfm.

---

# apple computer

**Autorisierter Fachhändler:**

BÜROTECHNIK
TEL. TECHNIK
COMPUTER
**dinkelbach**

Welfenhof · 33 Braunschweig · Tel. 05 31/4 01 67-69

---

# Preissenkung bei TEAC

| | |
|---|---|
| **FD 55 A,** 1 x 40 Track | **550,-** |
| **FD 55 B,** 2 x 40 Track | **631,-** |
| **FD 55 E,** 1 x 80 Track | **595,-** |
| **FD 55 F,** 2 x 80 Track | **695,-** |
| **FD 55 G,** 8" kompatibel | **810,-** |
| **Autodisk-Controler** (keine Patch-Disk erforderlich!) | **298,-** |
| **FDC 4,** für alle Laufwerke | **230,-** |
| **Patch-Disk,** | **80,-** |

**Ueding electronics**
Holtewiese 2
5750 Menden 1
DFÜ 02373/66877
Tel. 02373/63159

---

# apple computer

**Ihr Vertragshändler:**

BERATUNG
VERKAUF
HARDWARE
SOFTWARE
PERIPHERIE

**SCHÖPP GMBH**
TELECOMMUNICATION
6140 BENSHEIM · BAHNSTR. 79
TEL. 06251 · 39122

---

# apple

Der Vielseitige
**Apple II E 64 KB**
**DM 2.298,–**

Der Tragbare
**Apple II C 128 KB**
80 Zeichen, Disk u.a.
**DM 3.298,–**

Der Schnelle
**Apple Matrixdrucker**
„Imagewriter"
180 Z/Sek - Graphik
**DM 1.398,–**

Die Zuverlässige Original
**apple Disk**
**DM 848,–**

Die Flache
**Slimline Disk** mit Original apple Controler, Diskette und deutschem Handbuch
**DM 848,–**

Die Schöne
**Brother Typenraddrucker**
HR 15
**DM 1.398,–**

Die Sicheren Disketten Double Density in Zehnerpak.-Hardbox
100 Stk **DM 548,–**

Hard- und Software-Restposten günstig

Paketpreise noch günstiger

Das umfangreichste PC-Hardware-Zubehör gibt es für den **apple**
Info anfordern

Frei Haus, inkl. Mwst
1 Jahr Garantie - Rückgaberecht innerhalb von 8 Tagen

Hamburger Computer Versand
☎ 040 792 82 26
Postfach 610 125

---

# r+r electronic

Elektronik · Fachliteratur · Personal-Computer

## Qualitäts-Disketten 1.Wahl

**BASF-Disketten** softsektoriert in Pappkarton
5,25"-Disketten alle **ohne Aufpreis** mit Verstärkungsring

| Type | Beschreibung | 10 Stk | 100 Stk |
|---|---|---|---|
| 1XV | 48 TPI, 1-seitig, einfache Dichte | 47,90 | 469,00 |
| 1DV | 48 TPI, 1-seitig, doppelte Dichte | 49,50 | 484,50 |
| 1/96V | 96 TPI, 1-seitig, doppelte Dichte | 68,50 | 667,00 |
| 2/96V | 96 TPI, 2-seitig, doppelte Dichte | 82,50 | 799,50 |
| Aufpreis | für 10er-Pack in Hart-PVC-Klappbox | | je 5,00 |

**3M/Scotch Sicherheits-Disketten**
5,25"-Disketten **mit Verstärkungsring**; softsektoriert

| | | | |
|---|---|---|---|
| 744-0 | 48 TPI, 1-seitig, doppelte Dichte | 55,00 | 525,00 |
| 745-0 | 48 TPI, 2-seitig, doppelte Dichte | 75,90 | 712,00 |
| 746-0 | 96 TPI, 1-seitig, doppelte Dichte | 82,50 | 775,00 |
| 747-0 | 96 TPI, 2-seitig, doppelte Dichte | 95,50 | 896,00 |

**Wichtig** für die sichere Funktion Ihrer Disketten-Laufwerke
7440 Kopf-Reinigungsset für 5,25"-Laufwerke 49,50

**3,5" - Micro-Disketten für LISA und MACINTOSH** **NEU**
3,5"-Disketten 1-seitig, doppelte Dichte im 5er-Pack 74,50

**Plastikbox für 5,25" - Disketten**
Formschöne Klappbox aus Hart-PVC für ca. 10 Disketten in den Farben grün, orange, schwarz, hellgrau, elfenbein oder anthrazit. Bitte die gewünschte Farbe angeben. **Mischbar.**
**Deutsche Spitzenqualität 1 Stk . 6.95 10 Stk . 59.50**

**5,25"-Diskettenbox** **Sehr preiswert!**
Repräsentative Disketten-Box aus deutscher Fertigung in den Abmessungen (T x B x H) 306x174x144 mm ohne Schloß mit topas farbenem Deckel für ca. 80 Disketten und 4 beschriftbaren Stützplatten. **1 Stk ... 24.95** **ab 10 Stk ... 22.95**

**ABA - Diskettentröge**
Kompakter Trog **mit rauchfarbenem Deckel und Schloß.** Das Unterteil ist platzsparend im Deckel abstellbar. Je nach Größe incl. 4 bzw. 9 beschriftbaren Stützplatten.

| 5,25"-Mini-Disketten | Abm.: (T x B x H) | 1 Stk | ab 5 Stk |
|---|---|---|---|
| M35 | Abm.: 210x100x175mm für 40 Disk. | 39,90 | 36,85 |
| M85 | Abm.: 310x100x175mm für 90 Disk. | 58,50 | 51,85 |
| **8"-Standard-Disketten** | | **1 Stk** | **ab 5 Stk** |
| F40 | Abm.: 210x180x175mm für 40 Disk. | 61,70 | 55,50 |
| F90 | Abm.: 350x180x175mm für 90 Disk. | 87,60 | 78,85 |

## Datenkassetten
Universal * 1.Wahl
Geeignet für alle gängigen Home - Computer

| Bandmaterial: **Eisenoxid** | | | Bandmaterial: **Chromdioxid** | | |
|---|---|---|---|---|---|
| Best.-Nr | 10 St. | 50 St. 100 St. | Best.-Nr | 10 St. | 50 St. 100 St. |
| C10-FE | 15,95 | 74,80 142,50 | C10-CR | 17,95 | 85,00 162,50 |
| C20-FE | 16,95 | 79,50 152,50 | C20-CR | 19,50 | 92,00 175,50 |
| C30-FE | 18,65 | 87,50 167,50 | C30-CR | 22,25 | 105,00 199,50 |

Hochwertiges Band aus deutscher Fertigung. Jede CC in einer aufklappbaren Schutzbox mit je 2 Blanco-Aufkleber je Box.

**Apple - Zubehör**
**Unser großer Hit auch für Ihren Apple //e:**
**80 Zeichen/64kB-RAM für APPLE //e**
dtsch. Qualitätsprodukt
incl.Demo-Diskette **290.-**
Wichtig: Läuft 100% unter DOS, ProDOS, CP/M 2.2, CP/M 3.0 und PASCAL incl. Demo-Diskette und ausführlichem deutschen Handbuch.

**Z80B/64kB-Karte** (CP/M3.0,CBASIC,GSX80) 1195,00
**Original ALS incl. sämtlicher Software.**
**CP/M-PLUS-Manuals** (CP/M3.0) 75,00
Kompletter Satz bestehend aus: User's Guide, SID, System Guide, Programmer's Utilities- und Programmer's Guide.
**ITOH/EPSON-Vollgrafikinterface** m.Kabel 230,00
**Serielles Interface für APPLE Imagewriter** 235,00
**RS-232C - Karte** (Serielle Schnittstelle) 175,00
**Z-80 Karte** (ohne Software) Importkarte 149,00
**Z-80 Karte** (ohne Software) deutsche Herstellung 230,00
**Disk-Controller** (ohne Software) 4 Betriebsarten 195,00

**DISTAR-Laufwerk**
voll APPLE kompatibel
**495.-**
halbspurfähig, Spur-0-Erkennung, 40 Spuren, 143 kB.
Einzeln getestet für DOS, ProDOS, CP/M und PASCAL.
**Original APPLE PASCAL-Manuals** 75,00
(Satz=Language- und Operating System Reference Manual)

Der aktuelle Hinweis:
**SOFTWARE/HARDWARE - Gelegenheiten**
AIM/KIM/SYM/SUPERBOARD/NASCOM/SORCERER/APPLE u.a.
Aktuelle Preisliste unserer Gelegenheiten gegen Freiumschlag anfordern.

**Unverbindliche Vorführung und Beratung von sämtlicher Apple Hard- und Software.**

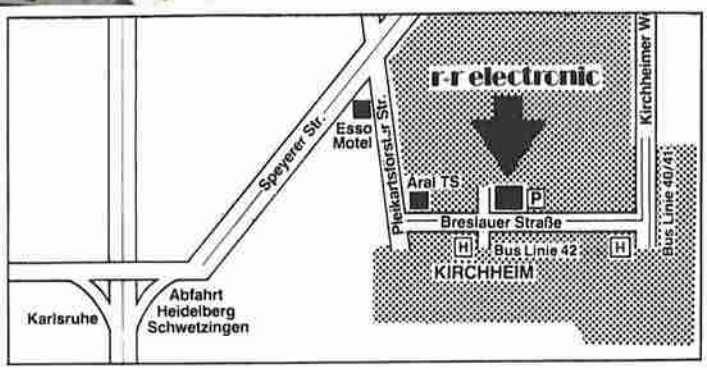
**6900 Heidelberg 1 Breslauerstr.29** Tel.06221/781500
**Geschäftszeiten:** Mo. - Fr 9-13 +14 - 18 Sa. 9 - 13 Uhr
**Versandanschrift:** 6900 Heidelberg 1 Dammweg 2

# r + r electronic

Der Heidelberger Apple-Vertragshändler r + r electronic Rufenach KG ist am 1. 1. 1977 in das Mikrocomputergeschäft eingestiegen, also genau zum rechten Zeitpunkt, als der „Apfelbaum" in Cupertino zu blühen anfing. Zunächst waren in dem Geschäft in der Adlerstraße auf 80 qm nur 3 Angestellte beschäftigt, deren Hauptverkaufsartikel anfangs elektronische Bauelemente, Meßgeräte usw. darstellten. Später kamen dann KIM, SYM, AIM-65, Nascom usw. hinzu, die an Industriefirmen und technische Institute sowie natürlich auch an Hardware-„Freaks" abgesetzt wurden. Übrigens hat auch der Chefredakteur von „Peeker" seinen ersten Apple – es war einer der ersten II+ Geräte – in dem seinerzeit noch räumlich sehr beengten Laden erworben.
Ursprünglich war r + r electronic ein „Ableger" der traditionsreichen Firma Arthur Rufenach Großhandlung für elektronische Bauelemente, die 1946 von dem Seniorchef gegründet wurde und heute allein 15 Angestellte im Dammweg umfaßt. Inzwischen hat sein Sohn, der Diplomphysiker Günther Rufenach, die Geschäfte übernommen.

Am 1.11.1981 erfolgte dann der Umzug der kleinen Einzelhandlung in die Breslauer Straße im Stadteil Kirchheim, der auch von auswärtigen Besuchern sehr günstig über die Autobahnabfahrt Schwetzingen erreicht werden kann. Auf über 200 qm reiner Ladenfläche (neben 200 weiteren qm Büro- und Lagerfläche) entstand nunmehr ein wirklich großzügig eingerichtetes Selbstbedienungsgeschäft, in dem allerdings auch die Beratung von Anfang an groß geschrieben wurde. Dort werden inzwischen 9 Mitarbeiter beschäftigt, die ein dreigeteiltes Verkaufsprogramm betreuen, das auch auf dem Postweg in die ganze Bundesrepublik verschickt wird:

– Elektronische Bauelemente und Bausätze aller Art: Zur Zeit sind ca. 8 000 verschiedene Bauteile ab Lager erhältlich. Das administrative Handling dieser Flut an „Kleinkram" erfolgt natürlich durch ein Apple IIe-Lagerprogramm, wobei allerdings hier applefremde Laufwerke mit größerer Datendichte herhalten müssen.
– Computerzeitschriften und -bücher: In den geschmackvoll eingerichteten Bücherregalen mit Frontalpräsentation findet man Hunderte von Büchern über Apple, EDV und Elektronik. Neben den deutschen sind auch die einschlägigen amerikanischen Periodika vertreten.
– Apple-Produkte aller Art vom Apple IIe über den Macintosh bis zum Lisa-Computer: r + r electronic ist ein Level 1 Service Center mit eigener Reparaturabteilung im 24-Stunden-Service. Neben Privatpersonen werden die Geräte insbesondere an Schulen, Universitäten und wissenschaftliche Institute verkauft.
An Apple-Software findet man außer den ureigenen Apple-Programmen wie Applewriter, Quickfile usw. vorwiegend kommerzielle Anwenderprogramme anderer Softwarehäuser, doch sind auch Utilities wie z.B. alle Programme des Hüthig Software Service erhältlich.
Abschließend sei nicht versäumt darauf hinzuweisen, daß sich das Verkaufspersonal, das ich schon seit Jahren kenne, durch die typische humorvolle „Heidelberger" Freundlichkeit auszeichnet, was ja im Zeitalter der Computer und Roboter nicht mehr immer eine Selbstverständlichkeit ist.



# Die Gewinner des Primzahlen-Wettbewerbs

Unser Primzahlen-Wettbewerb aus Heft 1/84 von „Peeker" ist mit exakt 279 eingesandten Lösungen auf eine ungewöhnliche Resonanz gestoßen, ungewöhnlich deshalb, weil die gestellte Aufgabe – in weniger als 0,45s alle Primzahlen bis 8191 zu errechnen – extrem schwierig und deshalb grundsätzlich nur in Assembler lösbar war. Zunächst ein herzliches Dankeschön an alle Teilnehmer, die sich sehr viel Mühe gegeben und hervorragende Lösungen eingereicht haben.

Als wir in der Redaktion die Aufgabenstellung für den Wettbewerb formulierten, rechneten wir zwar mit Einsendungen unter 0,2s, nicht jedoch mit seriösen Lösungen unter 0,1s, womit sich unser Rundungsverfahren für die 1024 Durchläufe als unzureichend erwies. Daher haben wir nachträglich den Kreis der Gewinner erweitert: Es gibt nebem dem 1. Preis *zwei* 2. Preise sowie über 30 Abonnement-Preise. Gewonnen habe alle, deren Programmdauer kleiner/gleich 0,08s ist.

Trotz dieser großzügigen Regelung mußten wir einige unseriöse Lösungen unter 0,08s disqualifizieren. So enthielt beispielsweise eine „Lösung" exakt 1028 STA-Befehle für genau diejenigen 1028 Primzahlen, die eigentlich hätten errechnet werden müssen. Bei einigen anderen ungültigen Lösungen wurde nach dem Piepston neu gebootet oder die Zero-Page war derart zerstört, daß es nicht mehr möglich war, die möglicherweise richtig errechneten Primzahlen im Speicher zu rekonstruieren. Schließlich wurden einige Lösungen erst nach dem 15.11.84 eingereicht.

Nunmehr wollen wir uns dem ersten Preis zuwenden, den Norbert Overmeyer aus Gelsenkirchen mit einer traumhaften Zeit von 0,050s gewonnen hat. Dieses Programm basiert auf einem Zahlensystem mit einer anderen Basis als 10 mit dem Vorteil, daß Mehrfachstreichungen von Primzahlen weitestgehend vermieden werden, womit sich die extrem geringe Laufzeit erklärt. Extrem gering ist auch der Umfang dieses Programms, das insgesamt nur ca. 250 Bytes einnimmt und sich deshalb von manchen „Holzhacker"-Lösungen mit teilweise bis zu 24K Umfang angenehm abhebt. Lassen wir nunmehr den Autor selbst zu Worte kommen:

Mein Name ist Norbert Overmeyer. Ich besuche den 13. Jahrgang der Gesamtschule Berger Feld in Gelsenkirchen. Meine Hobbys sind Informatik und Mathematik. Sie haben mich dazu gebracht, an diesem Preisausschreiben teilzunehmen. Ich habe ein Programm in Assembler geschrieben, das die Primzahlen zwischen 2 und 8191 in 0,050 Sekunden berechnet. Es folgt nun eine Beschreibung meines Lösungsprinzips. (Das Programm selbst befindet sich auf der Peeker-Sammeldiskette.)

Ich habe als Lösungsprinzip das Sieb des Eratosthenes verwendet, welches folgendermaßen arbeitet. Man schreibt die natürlichen Zahlen, die man auf Primzahlen hin untersuchen will, der Reihe nach auf. Nun werden nacheinander die Vielfachen der 2, 3, 4, 5 usw. gestrichen. Wenn man bei der letzten zu untersuchenden Zahl angelangt ist, sind nur noch Primzahlen übrig.

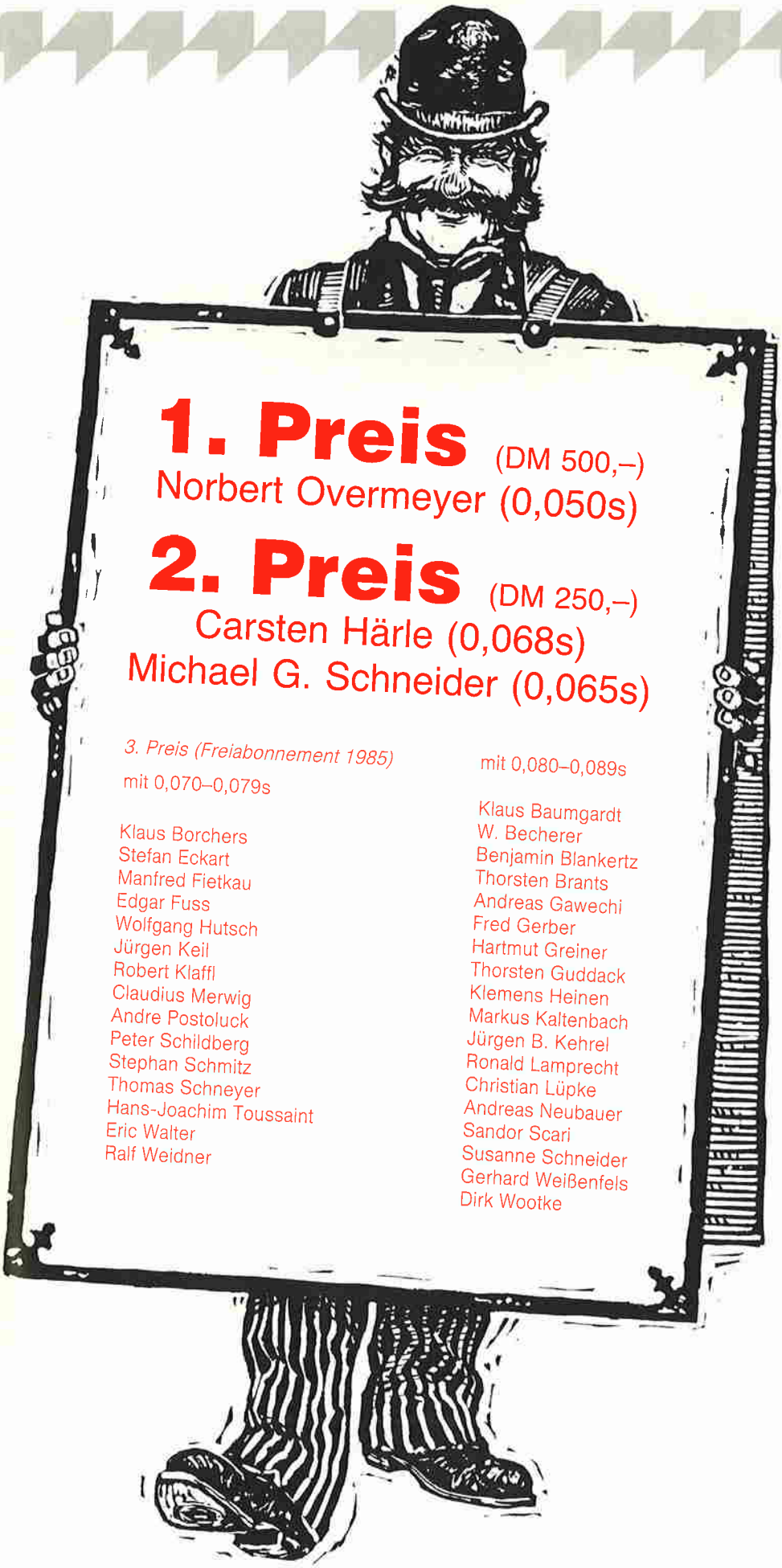
Bei dem oben erwähnten Verfahren werden viele Zahlen öfter gestrichen, z.B. die 6 von 2 und 3. Um das Verfahren zu beschleunigen, muß man nun nach Gesetzmäßigkeiten suchen, nach denen Nicht-Primzahlen nicht so oft gestrichen werden. Ich zähle nun die auf, die ich bei meinem Programm verwendet habe:

**1.** Ich brauche Vielfache von Nicht-Primzahlen nicht zu streichen, wenn sie bereits von einem Teiler dieser Zahl gestrichen wurden. Das heißt, (nur) wenn ich das Streichen aller Vielfachen einer Zahl auslassen will, muß ich zuerst feststellen, daß die Zahl selbst schon gestrichen wurde.

**2.** Wenn man von den Quadraten der Primzahlen absieht, haben alle Nicht-Prim-

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 × 30 = 0 | | | ○ | ○ | | ○ | | ○ | | | | ○ | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | ○ |
| 1 × 30 = 30 | | ○ | | | | | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | ○ | | | | | | ○ |
| 2 × 30 = 60 | | ○ | | | | | | ○ | | | | ○ | | ○ | | | | | | ○ | | | | ○ | | | | | | ○ |
| 3 × 30 = 90 | | | | | | | | ○ | | | | ○ | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | |
| 4 × 30 = 120 | | | | | | | | ○ | | | | ○ | | | | | | ○ | | ○ | | | | | | | | | | ○ |
| 5 × 30 = 150 | | ○ | | | | | | ○ | | | | | | ○ | | | | ○ | | | | | | ○ | | | | | | ○ |
| 6 × 30 = 180 | | ○ | | | | | | | | | | ○ | | ○ | | | | ○ | | ○ | | | | | | | | | | |
| 7 × 30 = 210 | | ○ | | | | | | | | | | | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | ○ |
| 8 × 30 = 240 | | ○ | | | | | | | | | | ○ | | | | | | ○ | | | | | | ○ | | | | | | ○ |
| 9 × 30 = 270 | | ○ | | | | | | ○ | | | | ○ | | ○ | | | | | | | | | | ○ | | | | | | |
| 10 × 30 = 300 | | | | | | | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | | | | | | | |
| 11 × 30 = 330 | | ○ | | | | | | ○ | | | | | | | | | | ○ | | ○ | | | | ○ | | | | | | ○ |
| 12 × 30 = 360 | | | | | | | | ○ | | | | | | ○ | | | | | | ○ | | | | ○ | | | | | | ○ |
| 13 × 30 = 390 | | | | | | | | ○ | | | | ○ | | | | | | | | ○ | | | | | | | | | | ○ |
| 14 × 30 = 420 | | ○ | | | | | | | | | | ○ | | ○ | | | | | | ○ | | | | ○ | | | | | | ○ |
| 15 × 30 = 450 | | | | | | | | ○ | | | | ○ | | ○ | | | | ○ | | | | | | | | | | | | ○ |
| 16 × 30 = 480 | | | | | | | | ○ | | | | ○ | | | | | | | | ○ | | | | ○ | | | | | | ○ |
| 17 × 30 = 510 | | | | | | | | | | | | ○ | | ○ | | | | | | | | | | | | | | | | |
| 18 × 30 = 540 | | ○ | | | | | | ○ | | | | | | | | | | ○ | | | | | | ○ | | | | | | ○ |
| 19 × 30 = 570 | | ○ | | | | | | ○ | | | | | | | | | | ○ | | | | | | ○ | | | | | | ○ |

**Tabelle 1:** Primzahlen zwischen 2 und 599.
Wenn Sie Zeilenwert und Spaltenwert addieren, erhalten Sie die Zahl. Ein Kreis heißt Primzahl, kein Kreis heißt keine Primzahl. Wenn man zwei Zahlen addiert, die beide durch eine Dritte teilbar sind, ist auch das Ergebnis durch diese Zahl teilbar. Da die Zeilenwerte durch 2, 3 und 5 teilbar sind, sind alle Spalten leer, deren Wert durch 2, 3 oder 5 teilbar ist.




zahlen mindestens 2 Teiler. Wenn ich nun beim Streichen des größeren der beiden Teiler angelangt bin, muß ich diese Zahl nicht mehr streichen, da sie schon durch den kleineren Teiler gestrichen wurde. Daher brauche ich erst beim Quadrat einer Zahl anfangen, seine Vielfachen zu streichen. Wenn dieses Quadrat dabei schon größer als die letzte zu untersuchende Zahl ist, sind alle Nicht-Primzahlen gestrichen und ich kann meine Arbeit beenden.

**3.** Sie wissen sicher, daß alle Zahlen, die als letzte Ziffer eine 0 oder 5 haben, durch 5 teilbar sind. Das liegt daran, daß die 10, die Basis unseres Zahlensystems, durch 5 teilbar ist. So kann man bei hexadezimalen Zahlen, z.B. $15 = 21, die Teilbarkeit durch 5 auf diese Weise nicht feststellen. Wenn ich nun mit einem Zahlensystem der Basis 30 = 2 * 3 * 5 rechnen würde, könnte ich anhand der letzten Ziffer feststellen, ob die Zahl durch 2, 3 oder 5 teilbar ist. Damit müssen alle Primzahlen – von 2, 3 und 5 abgesehen – auf 1, 7, 11, 13, 17, 19, 23 oder 29 enden. (Da dieses Zahlensystem nicht gebräuchlich ist, habe ich diese Endziffern dezimal angegeben.) So sind nur noch 8 von 30 Zahlen potentielle Primzahlen. Man kann, um sich das Prinzip zu verdeutlichen, die zu untersuchenden Zahlen auch in eine Tabelle eintragen, bei der in jeder Zeile 30 Zahlen stehen (s. **Tabelle 1**). Dabei wären 22 der 30 Spalten leer – 2, 3 und 5 ausgenommen. Ich habe die Basis 30 als Beispiel genannt, da hier das Ganze übersichtlicher ist. In meinem Programm habe ich mich für die 210 als Basis entschieden, da bei ihr noch die Vielfachen der 7 als mögliche Primzahlen wegfallen. Ich brauche also 210 aufeinanderfolgende Zahlen, bei denen ich (nur) die Vielfachen der 2, 3, 5 und 7 streiche und die die Zahlen 2, 3, 5 und 7 selbst nicht enthalten, z.B. 8 bis 217. Diesen Bereich kann ich dann nach 218 bis 427 und nach 428 bis 637 usw. kopieren. Wenn Sie nun meinen, ich setze die 2, 3, 5 und 7 als Primzahlen voraus, möchte ich Ihnen folgendes entgegnen: Ich kenne keine Primzahl, sondern nur die Zahl 210, und ich weiß nicht, daß 2, 3, 5 und 7 Primzahlen sind, sondern nur, daß sie Teiler der 210 sind. Ich erhalte auch das richtige Ergebnis, wenn ich mit der Basis 2 * 3 * 4 * 5 = 120 rechne und anstatt der Vielfachen der 7 die Vielfachen der 4 streiche.

# Hinweise für Autoren

Wir suchen laufend Beiträge: Erfahrungsberichte über kommerzielle Hardware- und Software-Produkte, Tips und Tricks für Anwender und Programmierer, gut dokumentierte Programme für Anfänger und Fortgeschrittene, hardwaretechnische Beiträge usw. Wir zahlen DM 200,- pro Druckseite, gleichviel ob Text oder Programm, wobei ein Beitrag mehrere Druckseiten oder gar mehrere Aufsatzfolgen umfassen kann. Es lohnt sich also, für uns zu schreiben.
Alle Textbeiträge einschließlich der Programmlistings werden von einem Apple IIe direkt in die Lichtsatzanlage unserer Druckerei übertragen. Im einzelnen sind deshalb folgende Richtlinien zu beachten:

Beiträge müssen sowohl als Papierausdruck wie auch als Diskette eingereicht werden. Zur Zeit können ausschließlich die klassischen 51/4 Zoll Disketten für Apple II/IIe/IIc angenommen werden. Berücksichtigt werden grundsätzlich nur Originalaufsätze und Originalprogramme. Es ist jedoch urheberrechtlich zulässig, aus Werken anderer Urheber zu zitieren. So können Sie z.B. Teile aus einem fremden Programm in Ihr eigenes übernehmen. In diesem Fall muß jedoch an geeigneter Stelle ein Zitathinweis erfolgen, etwa in der Form „Zeilen X-Y enthalten den Teil Soundso aus dem Programm ABC des Autors X, erschienen bei Firma Y, Ort und Jahr".
Wie im Verlagswesen üblich, übertragen Sie uns mit den eingereichten Aufsätzen und/oder Programmen das Recht der Vervielfältigung und Verbreitung in jeder Form. Speziell in bezug auf die Programme besagt dies, daß sie von uns nicht nur in Druckform, sondern auch in Diskettenform vertrieben werden, und zwar als Peeker-Sammeldisketten, die in unregelmäßiger Folge erscheinen und eine Auswahl der größeren Programme als Quell- und Objekt-Code enthalten. Versehen Sie Ihre Programme nicht mit dem amerikanischen Copyright-Vermerk, sondern schreiben Sie statt dessen nur „Von Name, Ort und Jahr".

## 1. Textbeiträge

a) Die Aufsatztexte sollten im Idealfall mit einem DOS-3.3-Textprogramm erstellt werden, das normale ASCII-Textfiles erzeugt, z.B. Applewriter II/IIe usw. Wordstar-Dateien im CP/M-Format sind ebenfalls möglich, doch müssen in diesem Fall die Texte endlos (ohne Silbentrennung) erfaßt werden.
b) Die Texte dürfen keinerlei Steuerzeichen für Matrixdrucker usw. enthalten. Überschriften dürfen nicht unterstrichen werden. Vielmehr brauchen lediglich im ausgedruckten Manuskript die Überschriften sowie darüber hinaus im Text selbst Hervorhebungen mit einem Farbstift o.ä. markiert zu werden. Die entsprechende Kodierung für die Satzanlage wird dann von uns übernommen.
c) Der Text muß grundsätzlich endlos erfaßt werden, d.h. ein Return darf nur am Absatzende und niemals am Zeilenende stehen, da der Satzrechner sonst keinen automatischen Umbruch mehr vornehmen kann. Vermeiden Sie darüber hinaus Einzüge zu Beginn eines Absatzes. Auch dies besorgt nämlich die Satzanlage automatisch.
d) Ausgangspunkt unserer Kodierung ist der deutsche ASCII-Zeichensatz, also mit Umlauten, ß usw. anstelle von eckigen und geschweiften Klammern. Wenn Sie – z.B. beim alten Apple II – keine Umlaute usw. eingeben können, rufen Sie uns an. Bei Texten, die *sowohl* die deutschen *als auch* die amerikanischen ASCII-Sonderzeichen enthalten sollen, z.B. bei Pascal-Aufsätzen, müssen die letzteren kodiert werden. Kodierung bedeutet, daß eine Buchstabenkombination, z.B. „§p" für griechisches π durch unser Transmitter-Programm in den entsprechenden Kode der Satzanlage konvertiert wird.
e) Vermeiden Sie unnötige Einrückungen und Kurztabellen. Ferner mischen Sie bitte nicht Aufsatztexte mit Programmtexten, da letztere gesondert in die Satzanlage übertragen werden. Größere Tabellen, die von uns konventionell gesetzt werden, sind auf gesonderten Textfiles zu speichern. Es genügt, wenn Sie senkrechte Trennlinien mit Kuli o.ä. andeuten. Schaubilder usw. sind stets willkommen. Hierzu genügen Bleistiftskizzen als Vorlagen für unsere Reinzeichner.
f) Achten Sie bitte auf Orthographie und Interpunktion. Deutsch ist immer noch eine wichtigere Sprache als Basic und Assembler!

## 2. Programmlistings

Beachten Sie, daß wir im Gegensatz zu anderen Computer-Zeitschriften Programmlistings nicht von den meist sehr häßlichen Printouts abphotographieren, sondern wie die Textbeiträge von Diskette in die Satzanlage übertragen, wobei eine besondere Schreibmaschinen-Lichtsatzschrift verwendet wird. Nur bei exotischen Programmiersprachen wählen wir z. Zt. noch den konventionellen Weg.

### 2.1. Basic-Programme

a) Applesoft-Programme usw. liefern Sie bitte als normale Programm-Files, die von uns für die Satzanlage aufbereitet werden.
b) Zeilennummern sollten stets dieselbe Stellenzahl haben, z.B. nur 3stellige Zeilennummern 100, 110, 120 o.ä., damit diese Zahlen von uns korrekt freigestellt werden können (RENUMBER anwenden).
c) REM's sollten der besseren Lesbarkeit halber möglichst in Groß-Klein-Buchstaben geschrieben werden.
d) Menü-Texte, REM's usw. müssen unbedingt in Deutsch sein. Englischsprachige Programme werden nicht angenommen. Englische Computer-Fachausdrücke sind demgegenüber selbstverständlich zulässig.

### 2.2. Assembler-Programme

a) Assembler-Programme sind als Source-Code und als Objekt-Code auf Diskette einzureichen. Wir bevorzugen den Assembler namens Merlin (= Big Mac) von G. Bredon. Verwendbar sind ferner Lisa 2.5, Toolkit-Assembler, ProDOS-Assembler sowie jeder andere Assembler, bei dem die Möglichkeit besteht, anstelle auf Papier auf Diskette zu assemblieren.
b) Der Source-Code sollte möglichst im 40-Z/Z-Modus geschrieben werden, damit die assemblierten Listings in der Zeitschrift 2spaltig gesetzt werden können. In der Kommentarspalte des Source-Codes sollten nur ganz kurze Bemerkungen stehen. Längere Kommentare schreibe man *über* die Befehle in reine Kommentarzeilen, wobei auch diese 40 Zeichen pro Zeile nicht überschreiten sollten.
c) Vermeiden Sie Makros und seltene Pseudo-Op-Codes, damit der Quell-Code ohne großen Aufwand in einen anderen Assembler konvertiert werden kann.

### 2.3. Sonstige Programme

Programme in selteneren Programmiersprachen sind willkommen, erfordern jedoch eine vorherige Absprache wegen der Konvertierungsmöglichkeit. Ggf. muß ein Schönschreibdrucker-Printout erstellt werden.


Für unsere Zeitschrift »**peeker**« suchen wir noch einen

**Redakteur**

Er sollte den Apple II in- und auswendig kennen, flott (mindestens 30 Baud), dudenfest und stilsicher schreiben können und vorzugsweise über CP/M- und Pascal-Programmierkenntnisse verfügen. Es erwartet ihn eine äußerst reizvolle Aufgabe in einem großen Verlagshaus in einer der schönsten Städte Deutschlands.

Ihre Bewerbung mit den üblichen Unterlagen sowie mit Proben Ihrer bisherigen schriftstellerischen Tätigkeiten richten Sie bitte an
**Verlagsgruppe Dr. Alfred Hüthig,** Personalabteilung,
**Postfach 10 28 69, 6900 Heidelberg 1**

**Hüthig**

# INFOSOFT

Vertriebsgesellschaft für Informationsverarbeitungs-Systeme und Software GmbH

Friedrich-Mohr-Str. 7 · 5400 Koblenz · Tel. (0261) 81068

- Computersysteme
- Branchenprogramme
- Beratung/Schulung
- Service

Programme für
- Fakturierung
- Lagerbuchhaltung
- Textverarbeitung
- Angebotswesen
- Fibu – Lohn und Gehalt
- Statistische Auswertungen
- Auftragsbearbeitung
- Kalkulation

apple computer

---

iBS

**Die besten Karten für Ihren Apple®**

## Alle Trümpfe aus einer Hand

Ein Textsystem wie im Film mit:
AP 17 256K RAM als Pseudodisk
Basic, Pascal, CP/M DM 1690,–
AP 22 Z80B CPU 6 MHz mit 64k RAM DM 895,–
Das Textprogramm Wordstar DM 1310,–

Bestechende Leistung!

| | |
|---|---|
| AP 20: 6800 CPU mit 128K RAM | DM 1459,– |
| AP 20 mit Assembler und Pseudodisk | DM 1595,– |
| AP 20 mit CP/M 68K | DM 2268,– |
| AP 26-A-256K RAM für AP 20 | DM 1695,– |
| AP 26-B-1 Megabyte für AP 20 | DM 6450,– |
| AP 20 + AP 26 + CP/M 68K | DM 3642,– |
| CP/M 68K mit C-Compiler | DM 1356,– |
| Fortran, Pascal, Basic u.a. | auf Anfrage |

Disketten mit Innenlochverstärkung ss, sd
10er Pack DM 55,–
100er Pack DM 500,–

Diese Kombination bestimmt Ihre Computerzukunft

| | |
|---|---|
| 16K RAM Karte | DM 165,– |
| 80Z Karte mit Softswitch | DM 295,– |
| Z80A Karte | DM 165,– |
| zusammen nur | DM 595,– |

alle Preise incl. MWSt.

iBS COMPUTERTECHNIK
Olper Str. 10
4800 Bielefeld 14
Tel. 0521/444032
West-Germany

1011 Rose Marie Lane 16
Stockton. CA 95207
Tel. (209) 473-7473
USA

---

# NEU

Dipl.-Ing. R. Springmann
Stöckener Straße 199 · 3000 Hannover 21
Tel. (0511) 791111 · Telex 921466 comps d

## DISTAR-Laufwerke f. APPLE-II und ähnliche

- APPLE-II kompatibel
- halbspurfähig
- 40 Spuren, 163 KB
- Spur-0-Erkennung
- Direktantrieb
- Stahlband-Positionierung
- Kabel f. DISK-II-Controller
- kann DOS 3.3, Pascal, CP/M

**498,–** (inkl. MwSt.)

# Disketten

Markenfabrikat "Sentinel" 5.25". SS/DD/40 Spuren, mit Verstärkungsring

| | |
|---|---|
| 10 Stück | DM 49,–– |
| 50 „ | DM 230,–– |
| 100 „ | DM 450,–– |

**Händler-Anfragen erwünscht!**

---

**Qualität muß doch nicht teuer sein!!** Peripheriegeräte für Apple u. Kompatible zu vernünftigen Preisen.

**Die absolute Preissensation!!!!** SIEMENS Disklaufwerk F122 im Gehäuse, anschlußfertig **498,–**

| | | | |
|---|---|---|---|
| STAR Drucker Delta 10 | 1295,– | STAR Drucker Gemini 10X | 898,– |
| PHILIPS Monitor TP 200 | 225,– | PHILIPS Monitor V7001 | 315,– |
| Z-80 Karte, 16KB RAM | 109,– | 80-Zeichenkarte m. Softw. | 189,– |
| Diskcontroller | 109,– | Druckerinterface m. Kabel | 169,– |
| 128 KB RAM (Saturn Komp.) | 549,– | 8er. Interface (CCS Komp.) | 199,– |
| DISTAR-Slimline, 163 KB | 485,– | Autopatchcontroller | 325,– |

Disksubsystem für APPLE II u. IIe
2 x 80 Track Laufwerke, Netzteil, Controller, Gehäuse, Handbuch keine Patchsoftware mehr erforderlich, für alle Betriebssysteme

**Einführungspreis 2495,–**

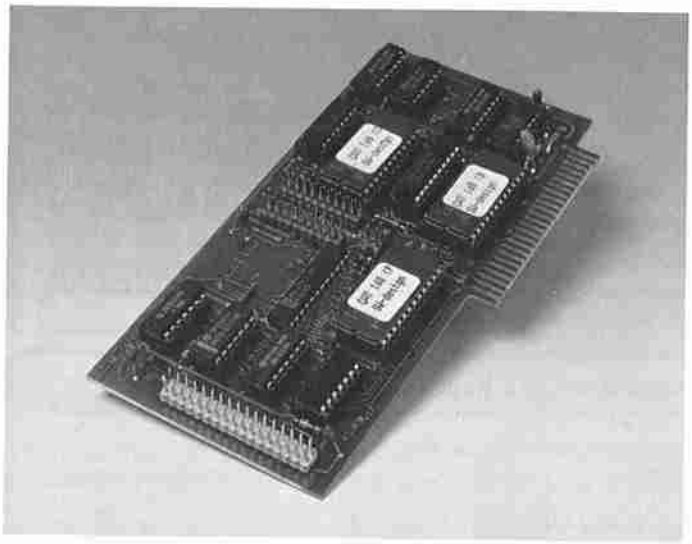
Am besten Sie rufen gleich an.

**TECHNOCONSULT GmbH** Wiedehagen 18, 4400 Münster, Telefon 0251-719116 u. 717420, Mo–Fr 9–17 Uhr

# Diskettenlaufwerke für ProDOS im Test

Aus der Sicht der Dateiverwaltung eignet sich ProDOS theoretisch besser für größere Laufwerke als für die klassischen 140K-DISK-II-Drives. Allerdings hat sich die Firma Apple einige „Tricks" ausgedacht, die die Anpassung von ProDOS an andere Laufwerke und Controller unnötigt behindert. Die Peeker-Redaktion hatte Gelegenheit, mehrere der für den Apple II gedachten Laufwerke zu testen, zu denen nachfolgend stichwortartig einige Erfahrungen mitgeteilt werden sollen, stichwortartig deshalb, weil die dazugehörigen Patch-Utilities teilweise noch nicht vorliegen. Im einzelnen wurden folgende Drives untersucht, wobei auf Typ, Installation, Verschlußmechanismus (Laufwerkklappe), Datenübertragungsrate (Geschwindigkeit) und insbesondere auf ProDOS-Kompatibilität geachtet wurde. Für die Geschwindigkeitstests wurde ein 32K langer Binärfile je 10mal gespeichert und eingelesen (10mal BSAVE TEST, A$1000, L$7FF0 und 10mal BLOAD TEST).

– Duodisk-Laufwerke (Fa. Apple, München)
– Distar-Laufwerke (Alleinvertrieb durch Fa. Springmann, Stöckener Str. 199, 3000 Hannover 21, Tel. 0511/791111)
– Sony-Laufwerke mit speziellem Controller (letzterer von Fa. Ueding electronics, Holtewiese 2, 5750 Menden, 02373/63159)
– Mitsubishi-Laufwerke mit speziellem Controller (letzterer von Fa. Stolte-Wilken, Iternberg 86, 5100 Aachen-Kor., Tel. 02408/4249)



**Duodisk-Laufwerke**

*Typ:* Doppel-Laufwerke für 5,25-Zoll-Disketten mit 35 Spuren, einseitig, einfache Dichte. Theoretisch kann man noch die 36. Spur formatieren, doch sind die Laufwerke hierfür nicht ausgelegt. Der Controller hat keine getrennten Anschlüsse für beide Drives. Deshalb kann der Controller nicht für die alten und weiterhin lieferbaren DISK-II-Laufwerke verwendet werden.

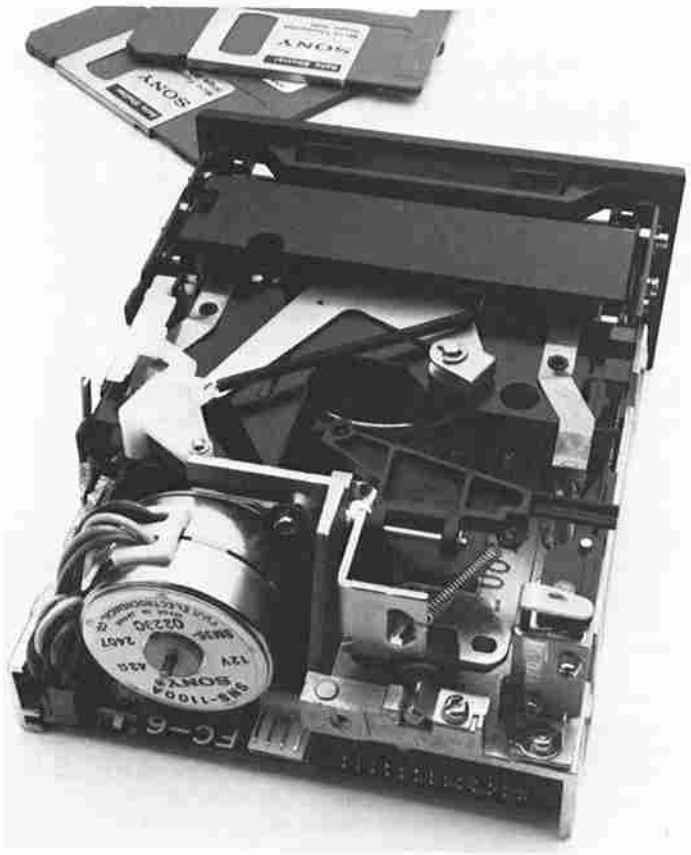
*Installation:* unkompliziert. Wenn jedoch eines der beiden Drives defekt ist – wie bei uns in der Peeker-Redaktion geschehen – so muß man die Duodisk komplett zur Reparatur geben. Würde ein Händler keinen zwischenzeitlichen Ersatz stellen, hätte man vorübergehend gar kein Laufwerk.

*Verschluß:* schlecht gelöst. Bei Einlegen der Diskette spürt man keinen „Anschlag", so daß man nie genau weiß, ob sie bereits richtig eingelegt ist oder nicht. Bei den DISK-II-Drives konnte man die Diskette zentrieren, indem man beim Einschalten des Motors die Laufwerkklappe behutsam schloß. Dies ist bei den Duodisk-Drives nicht mehr möglich. Beim Vordrükken des Verschlusses springt die Diskette nach vorne, wobei man aufpassen muß, daß man die Diskette nicht mit dem Daumen biegt. Das typische „Rappeln" der DISK-II-Drives ist bei den Duodisks fast noch lauter.

*Datenübertragungsrate:* bei DOS 3.3 946 Bytes/s, bei ProDOS 2505 Bytes/s, bei Diversi-DOS 2-C 4870 Bytes/s

*ProDOS-Kompatibilität:* kein Problem

*Fazit:* für kleine Datenmengen ausreichend



**Distar-Laufwerke**

*Typ:* Einfach-Laufwerke, die an die normalen DISK-II-Controller angeschlossen werden können (maximal 2 je Controller) für 35- bis 40-Spur-Disketten, 5,25 Zoll, einseitig, einfache Dichte.

*Installation:* unkompliziert

*Verschluß:* gut gelöst, da Hebelverschluß. Zentrierungs-I/O-Fehler treten praktisch nie auf; kein „Rappeln", sehr laufruhig.

*Datenübertragungsrate:* wie bei den Duodisks

*ProDOS-Kompatibilität:* kein Problem. Für 40-Spur-Formatierung kann unser PRODOS.INIT-Programm verwendet werden.

*Fazit:* für kleinere Datenmengen ausreichend. Preiswerter als Duodisk-Laufwerke.

**Sony-Laufwerke**

*Typ:* 3,5-Zoll-Laufwerke von Sony Communication Systems, 80 Spuren (320K), einseitig (doppelseitige Laufwerke in Vorbereitung). Controller der Firma Ueding mit Patch-Disketten für „getuntes" DOS 3.3, Pascal usw., jedoch noch nicht für ProDOS.

*Installation:* bei dem uns zur Verfügung gestellten Prototyp etwas umständlich, doch soll sich dies ändern.

*Verschluß:* sehr gut gelöst. Im Gegensatz zum Macintosh-Drive mit Auswurfknopf. (Beim Macintosh muß man, wenn ein geschütztes Programm abstürzt und die Resettaste nicht installiert ist, eine Büroklammer nehmen!)

*Datenübertragungsrate:* bei ProDOS 2829 Bytes/s, bei „getuntem" Ueding-DOS 3.3 5240 Bytes/s, bei Diversi-DOS 4502 Bytes/s

*ProDOS-Kompatibilität:* In der „Peeker"-Redaktion haben wir uns „auf die Schnelle" diese Laufwerke auf ProDOS „hochgezogen". Zu diesem Zweck wurde zunächst eine Sony-Leerdiskette mit unserem PRODOS.INIT formatiert. Dann wurden manuell für Slot 5 folgende Patches vorgenommen:

CALL -151

BF35: 50
BF31: 03
BF1A: 00F8
Danach liefen die Sony-Laufwerke problemlos unter ProDOS, so daß entsprechende Geschwindigkeitstests durchgeführt werden konnten. Allerdings kann bei dem momentan in Überarbeitung befindlichen Controller noch kein ProDOS kalt gebootet werden, weil das Controller-EPROM $CsFF: FF enthält, wodurch Apple III vorgetäuscht wird.
*Fazit:* für größere Datenmengen zu empfehlen.

**Mitsubishi-Laufwerke**
*Typ:* 5,25-Zoll-Doppel-Laufwerke von Mitsubishi Electric, doppelseitig, doppelte Dichte, damit 4mal 80 Spuren. Mit speziellem Controller CAT 160 der Firma Stolte-Wilken, der auch für Teac-, Shugart- und YE-Data-Laufwerke verwendet werden kann. Patch-Disketten für DOS 3.7 (Phantasie-Nr. ?), Pascal usw. lieferbar, jedoch noch nicht für ProDOS.
*Installation:* problemlos
*Verschluß:* weniger komfortabel als bei den Distar- und Sony-Drives, jedoch besser als bei den Duodisk-Laufwerken.

*Datenübertragungsrate:* bei DOS 3.7 1517 Bytes/s; ProDOS und Diversi-DOS funktionierten nicht. Neben „DOS 3.7" gibt es auch noch ein schnelles DOS 3.3, das uns allerdings nicht vorlag.
*ProDOS-Kompatibilität:* Bei dem CAT 160 müßte nicht nur ein eigenes Formatierungsprogramm geschrieben werden, sondern auch ProDOS – insbesondere wegen der beidseitigen Benutzung der Disketten – intensiv gepatcht werden. Zum Vergleich ist bei den Sony-Laufwerken praktisch gar kein Patch erforderlich.
*Fazit:* für große Datenmengen geeignet, allerdings im Moment nicht unter ProDOS.

**Gesamtfazit:** Die Sony-Laufwerke haben uns am besten gefallen, weil sie sehr klein, handlich und kompakt sind, weil sie mit 35, 40 und 80 Spuren formatiert werden können, weil auch normales DOS 3.3 sowie häufig benutzte Varianten wie Diversi-DOS ungepatcht benutzt werden können und weil für ProDOS keine aufwendige Modifikation erforderlich ist.

## Atari-Joystick und Apple

Der von der Firma Tombstone-Micro vertriebene Joyport ermöglicht den Anschluß von zwei Atari-, Commodore- oder Spectravideo-Joysticks an den Apple-Gameport. Dieses Produkt ist für Applebenutzer gedacht, die einen oder zwei Atari-Joysticks an ihren Computer anschließen möchten. Im Gegensatz zu den Original-Joysticks besitzen diese keine Potentiometer, sondern nur vier Schalter. Sie geben die Signale also digital aus. Auf der Platine werden die Signale des Atari-Joysticks in applespezifische (analoge) umgewandelt. Dabei ersetzt Joystick 1 die Apple-Paddle 0 und 1, Joystick 2 Paddle 2 und 3.
Viele Spiele sind auf diese Art von Joysticks eingerichtet, z.B. Pacman (und seine Varianten), Twerps, Donky Kong, Bandits, Space Raiders und viele andere.
Die Installation des Joyports ist schnell getan. Das ca. 30 cm lange Flachbandkabel wird wie ein gewöhnlicher Applejoystick in den Gameport eingesteckt. Am anderen Ende des Kabels befinden sich auf einer kleinen Platine zwei Anschlüsse, an diese können nun Atari-Joysticks (oder kompatible) angeschlossen werden. Danach kann die Funktion mit einem kurzen Applesoftprogramm, das sich auf dem Beilageblatt befindet, getestet werden.
Der Preis des Joyports beträgt DM 40,-. Zum Lieferumfang gehören die Platine mit Kabel und eine einseitige Beschreibung (die jedoch vollkommen ausreichend ist).
Bezugsquelle: Tombstone-Micro, Gardeschützenweg 72, 1000 Berlin 72, Tel. 030/8331303

## MacLabel

Bei MacLabel, einem Programm im Macintosh-Design, handelt es sich um eine neue Adreßverwaltung.
Dieses Paket ist im Gegensatz zu anderen Programmen genau auf den Macintosh zugeschnitten. Das heißt, es nutzt die Steuerung mit der Maus und die enormen Grafikmöglichkeiten des Macintosh. Fenstertechnik, Nutzung aller anschließbarer Peripherie und Zusammenwirken mit anderer Macintosh-Software, wie zum Beispiel MacWrite, sowie Ablauffähigkeit auf der LISA. Alle Funktionen des Macintosh, wie Ausschneiden, Einsetzen und Kopieren, Drucken in verschiedenen Schriftarten, die der Macintosh zur Verfügung stellt, werden voll unterstützt.
Eine weitere Besonderheit findet man in den ausgedehnten Selektionsmöglichkeiten. Jede Adresse kann nach fast 100 Kriterien eingeteilt werden.
Andere Leistungsmerkmale sind, daß die Adressen komprimiert abgespeichert werden, d.h. ohne überflüssige Leerzeichen. Somit lassen sich auf einer Standard-Macintosh-Diskette bis zu 4000 Adressen abspeichern, was auch den Bedarf eines größeren Unternehmens deckt.
Preis DM 548.- inkl. MwSt.
Ein funktionsähnliches Programm namens Label II ist für die Familie des Apple-II-Computers zum Preis von DM 498,- erhältlich. Ein Test dieser letzteren Version ist für „Peeker" geplant.
Bezugsquelle: IWT Software Service GmbH, Dahlienstraße 4, 8011 Vaterstetten, Tel. 08106/31017

## CTS-Mineralöl

Für den Mineralöl- und Brennstoffhandel, wie er überwiegend als Klein- und Mittelbetrieb geführt wird, entwickelte CTS in Zusammenarbeit mit Anwendern, Steuerberatern und Zoll das Programmpaket MINERALÖL. Es umfaßt die vollständige Auftragsabwicklung mit Kundenverwaltung, Artikelverwaltung, Auftragsverwaltung, Lagerverwaltung, Fakturierung, Offene Posten, Lastschrifteinzug, Zahlungseingang und Mahnwesen.
Ein ausführlicher Statistikteil druckt zudem alle wesentlichen Informationen über Erlöse, Erfolgsrechnung, Kundenaufträge, Artikel- und Lagerumsatz auf Abruf aktuell aus.
Bezugsquelle: CTS Computer und Textsysteme Burger und Herr GmbH & Co. KG, Mathystraße 28, 7500 Karlsruhe 1, Tel. 0721/811035

### Die erste Peeker-Sammeldiskette ist da!

Nach jedem zweiten Heft von Peeker erscheint eine Sammeldiskette, die alle Programme der beiden vorangegangenen Hefte enthält, oft mit zusätzlichen Utilities, die in den Heften selbst aus Platzgründen nicht gelistet werden konnten.

Einzelpreis DM 28,–. Bei Fortsetzungsbezug DM 20,– pro Diskette. Jederzeit kündbar. Mindestbezug 6 Disketten. (Mindestbestellmenge für Händler 10 Exemplare).

Soeben erschienen: Sammeldiskette für Heft 1 und 2, 1984. DOS 3.3 Diskette, 35 Spuren, broschiert für Buchregal. Inhaltsverzeichnis:

A 002 PEEKER === HEFT 1/84 ===
A 002 A–
T 029 T.DISASSEMBLER.65C02
B 004 DISASSEMBLER.65C02
A 002 B–
T 004 T.ACCEL.WAIT
B 002 ACCEL.WAIT
T 008 T.ACCEL.BOOT
B 002 ACCEL.BOOT
A 002 ACCEL.LC.KOPIERER
T 008 T.ACCEL.LC.KOPIE
B 002 ACCEL.LC.KOPIE
T 010 T.ACCEL.ROM.KOPIE 1
B 002 ACCEL.ROM.KOPIE 1
T 008 T.ACCEL.ROM.KOPIE 2
B 002 ACCEL.ROM.KOPIE 2
A 002 C–
A 027 TURTLE GRAFIK MIT REMS
A 018 TURTLE GRAFIK OHNE REMS
A 002 D–
A 003 DOUBLE.LORES.SOFTSWITCH. DEMO
A 004 DOUBLE.LORES.APPLESOFT. DEMO
A 003 AMPER.DOUBLE.LORES.DEMO
T 009 T.AMPER.DOUBLE.LORES
B 002 AMPER.DOUBLE.LORES
T 011 T.DOUBLE.LORES
B 002 DOUBLE.LORES
A 002 E–
A 018 HIRES
T 040 T.PRINTHIRES
B 005 PRINTHIRES
A 002 PEEKER === HEFT 2/84 ===
A 002 F–
A 007 DHGR.APSOFT.DEMO
A 003 AMPER.DOUBLE.HIRES.BAS
B 004 AMPER.DOUBLE.HIRES
T 015 T.AMPER.DOUBLE.HIRES
A 008 DHGR.LINEPLOTTER
A 002 G–
A 003 INSTRING.TEST
B 002 INSTRING.OBJ
T 019 T.INSTRING.OBJ
B 014 INSTRING.LISA.SOURCE
A 002 H–
A 004 LOESCHEN.EINES.ARRAYS
A 002 I–
B 018 ULTRATERM.ENGLISCH
B 018 ULTRATERM.DEUTSCH
A 002 J–
A 004 PRIMZAHLEN.OVERMEYER
B 002 PRIM.OBJ0
B 002 PRIM.OBJ1
A 002 PRIM.TEST
T 012 PRIM.TOOLKIT.SOURCE

**Hüthig Software Service**
**Postfach 10 28 69 · 6900 Heidelberg 1**

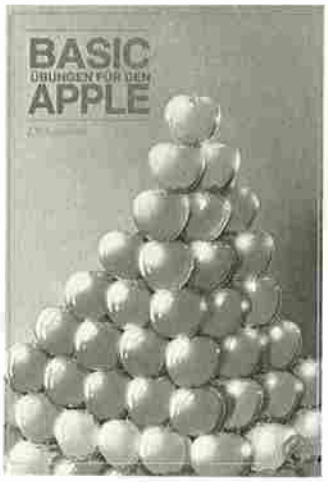
## BASIC Übungen für den Apple

von J. P. Lamoitier
1983, 252 S., zahlr. Abb., kart., DM 38,-
ISBN 3-88745-016-7
Sybex-Verlag, Düsseldorf

Das Buch ist konzipiert, allen Apple-Anwendern Applesoft-BASIC durch praktische Übungen an Hand von reellen Programmen beizubringen. Datenverarbeitung, Statistik, kommerzielle Programme, Spiele und vieles mehr. Jede Übung beinhaltet eine Beschreibung der Problemstellung, eine Analyse der Lösungsmöglichkeiten, ein Flußdiagramm und ein fertiges Programm samt Probelauf.

### Aus dem Inhalt:

Ihr erstes BASIC-Programm – Flußdiagramme – Übungen mit Integerzahlen – Elementare Beispiele aus der Geometrie – Allgemeine Übungen aus der Datenverarbeitung – Mathematische Berechnungen – Kaufmännische Berechnungen – Spiele – Operations Research – Statistik




### Neuer Buchkatalog

Der neue Katalog „Bücher für Electronic + Computer" der Firma Hamburger Buchversand GmbH, Fischertwiete 1, 2000 Hamburg 1, Tel. 040/330971, enthält auf 160 Seiten im DIN A 4-Format wahrscheinlich weit über 1000 Buchtitel aller deutschen Verlage, wobei Electronic-Bücher schwergewichtig vertreten sind. Allerdings findet man auch eine Vielzahl von Apple-Titeln.

### Schulsoftware-Katalog

Im Gegensatz zu dem obigen Pandasoft-Katalog ist der Schulsoftware-Katalog der Firma Klett in Stuttgart nur gegen eine Schutzgebühr erhältlich.
Grund: Die aufgeführten Programme können direkt von den mit kompletter Anschrift genannten Anbietern bezogen werden. Der Katalog gliedert sich in die Gebiete Informatik, Mathematik, Physik und Verwaltung.
Bezugsquelle: Ernst Klett Verlag, Kennwort: Apple-Katalog, Postfach 8 09, 7000 Stuttgart 1

## The Apple in your hand

von Ekkehard Flögel
1983, 220 S., zahlr. Abb. und Tabellen, kart., DM 39,-
ISBN 3-88963-178-9
Hofacker, Holzkirchen

Dieses Buch enthält eine Sammlung vermischter Programme in Applesoft, Assembler und Forth. Doch wer programmiert schon gleichzeitig in diesen drei Sprachen? Da keine Gegenüberstellung der einzelnen Programme erfolgt, wäre es sicherlich sinnvoller gewesen, wenn man jeweils einen gesonderten Band erstellt hätte. In der vorliegenden Form können wohl nur diejenigen Leser vollen Nutzen aus dem Buch ziehen, die ständig mit den drei Programmiersprachen arbeiten.

### Aus dem Inhalt (Auswahl):

Basic Section (Strings, Linked Lists, Hashing, Plotting Functions, Fourier Analysis) – Machine Language Section (Branches, Comparisons, Addressing, Command &, Input and Output with the 6522) – Forth Section (Fundamental Operations in Arithmetic, Constants and Variables, Access to the Disk, Defining Words, Low Resolution Graphics, High Resolution Graphics)

## Apple II Tips & Tricks

von Werner Voß und Renate Prust
1984, 405 S.,zahlr. Abb. und Tabellen, kart., DM 49,-
ISBN 3-89011-025-8
Data Becker, Düsseldorf

Dieses Buch ist im Grunde eine solide, umfangreiche Einführung in die Programmiersprache Applesoft. Leider wird durch den Werbetext auf dem Umschlag der Eindruck erweckt, als handele es sich hierbei um ein Werk mit vielen Insider-Tips und -Tricks. Davon kann jedoch nicht die Rede sein. Da wird z.B. von „Unterprogrammen in Maschinensprache" geredet. In Wahrheit scheinen die Autoren von Assembler nur wenig zu verstehen, sonst hätten sie hier mehr gebracht als ein einziges weniger als 10 Bytes langes Programm, das zu allem Überfluß auch noch aus einem anderen Werk zitiert wird. Ferner wird da z.B. von „Mischen von Texten und Graphik" geredet, und um offenbar auch wirklich jedem klarzumachen, daß dieses Thema behandelt wird, enthält der Klappentext dieses Stichwort gleich zweimal. Doch wenn man dann im Buch auf Seite 193 zu diesem Thema gelangt, wird man mit folgendem Satz abgespeist: „Dieses Problem kann auch mit geeigneten Graphik-Hilfsprogrammen gelöst werden, über die in diesem Buch aber nicht gesprochen werden soll. Der interessierte Leser wird entsprechende Informationen von jedem guten Software-Anbieter einholen können." Halten wir fest: Das Buch – das normalerweise keine schlechte Kritik verdient hätte – ist eine brauchbare Einführung in Applesoft-BASIC, die jedoch entgegen der von den Autoren möglicherweise selbst gar nicht intendierten Ankündigungen weder auf Spezifika des Apple IIe geschweige denn des Apple IIc eingeht noch Tips und Tricks für Fortgeschrittene bringt, so daß man Insider-Informationen vergeblich sucht.

### Aus dem Inhalt:

Die Nutzung peripherer Geräte – Speicherzugriff und Maschinenprogramme – Graphikprogramme – Verbesserung der Programmstrukturen – Editieren – Programmierhilfen – Alternativen zur BASIC-Programmierung

# Apple zu langsam?

Die TempoHexe ("SpeeDemon") macht den Apple II, II+ und IIe so schnell wie die populären 16bit Rechner. Die TempoHexe beschleunigt alle Programme bis zu 3 1/2 mal – absolut ohne jede Software-Änderung! Einfach einstecken, einschalten ... los.

Mehr Informationen und Händleranfragen bei:

**R. Alverdes**
**Schwarzwaldstr. 8a**
**7602 Oberkirch**
**Tel.: (0 78 02) 37 07**
**Telex: 752637 sfe d**

Katalog gegen DM 1,– in Briefmarken.

Wir haben die neuste Software und Peripherie für den Apple II, II+, IIe und Mac.

# ELECTRONIC-VERTRIEB LÄCHELE

CAD Hard- und Software · Vertrieb · Schulung · Beratung und eigener Service

**MITSUBISHI · IBM · EPSON · GENIE 16 · OLIVETTI · APPLE**

EDV-Zubehör, Interface-Karten und Hard Disk-Systeme für APPLE + IBM

**Saarbrücker Str. 7**
**Postfach 41 03 50**
**6800 MANNHEIM 31**
**Telefon 06 21 / 72 22 75 u.**
**06 21 / 73 90 08**

# Profi II

**der Apple-Kompatible!**

**64 K RAM,**
**2 CPUs (6502 + Z80A),**

**Profi II,** opt. wie Apple, inkl. 1 Floppy m. Controller **1998.–**
**Profi II** im Alu-Gehäuse, freistehende Tastatur m. Zehnerblock, 1 Floppy m. Controller ..... **2398.–**
**dito, mit 2 Laufwerken** ..... **3028.–**
**dito, mit 2 Laufwerken a. 320 K.** ..... **3385.–**
**dito, mit 2 Laufwerken a. 640 K.** ..... **3728.–**
Alle Rechner im Alu-Geh. werden mit ca. 300seit. deutschem Handbuch geliefert!
**Aufpreis für 7,5-A-Schaltnetzteil** ..... **50.–**
Weitere Modelle auf Anfrage!
**Erweiterungs-Karten,** z. B. Pio/Centr./AD-DA-Wandler usw. ..... **a. Anfrage**
**6 Monate Garantie!**

# Dorsch – electronic

**Forther Hauptstraße 23, 8501 Eckental 2**
**Telefon (09 126) 74 19**

# r+r electronic

**Elektronik · Fachliteratur · Personal-Computer**

## Neu im Lieferprogramm

**Problemlos mit Computern in aller Welt kommunizieren!** Mit dem Akustikkoppler AK300. Daten, Texte und Programme -per Telefon! VonDatenbanken, Großrechnern und Mailboxen, Zwischen Filialen, Außendienst und Zentrale, den Mitgliedern eines Clubs oder ganz privat ...

**Akustikkoppler AK 300** 548,00

**Technische Daten:** 300 baud, Vollduplex, Originate und Answer, Batterie- u. Netzbetrieb (Option), Variable Gummimuffen für alle Hörer, **FTZ-Nr. 18.13.1897.00., von der Post zugelassen. Gebühren- und anmeldefrei!**
Zum Anschluß an den Apple II+/IIe benötigen Sie keine Interfacekarte (Anschluß über den Game-Port) sondern unsere **Teleterm-Software** incl Kabel (nur Apple II+/IIe) ..... **198,00**
Oder Sie setzen für den Apple II+/IIe und den Apple IIc die sehr komfortable Terminal-Software von Apple ein ...

**Access//** Terminal Emulation Software 279,00

Access// ist das neue Terminal Emulation Programm von APPLE Computer für die Apple//-Familie.
Sie verwandeln Ihren Apple// in ein Terminal und können somit Daten von einem Host-Rechner oder Elektronik-Mail-System empfangen und versenden. Access// bietet weiterhin die Möglichkeit des Abspeicherns der zu empfangenden Informationen auf einer Diskette oder des Versenden von Dateien, die auf einer Diskette gespeichert sind. Access// basiert auf dem Betriebssystem ProDOS.

**jane...** das integrierte Softwarepaket 595,00

**jane** ist ein integriertes Sofwarepaket (Kalkulations- Textverarbeitungs-und Dateiverwaltungsprogramm) mit der Maus. Alle Tastaturkommandos wurden durch einfache, leicht zu verstehende Bilder ersetzt, über die man dem Computer sagt, was er tun soll. Die integrierte Hilfe-Funktion von **jane** erläutert alle Steuersymbole des Systems. Damit kann der Anfänger das Programm bedienen, ohne umfangreiche Handbücher zu lesen. (**ohne Maus bitte separat bestellen.**)

Apple//e-Maus mit Mouse-Paint ..... 545,00
Apple//c-Maus mit Mouse-Paint ..... 298,00

**AppleWorks** Integriertes Sofwarepaket 850,00

AppleWorks (AW) kombiniert die Textverarbeitung, eine Datenbank und ein Rechenblatt in einem integrierten Paket. AppleWorks ist anwenderfreundlich gestaltet. Ein "Schreibtischmanager" koordiniert die einzelnen Funktionen. Man kann gleichzeitig in verschiedenen Dokumenten arbeiten und in sekundenschnelle Informationen aus einem Dokument ausschneiden und in ein anderes einsetzen(**deutsche Version**).

**Deutsche Programme**

**ProDOS - Editor 1.0** ..... 98,00
Zeilenorientierter Editor für den APPLESOFT-Programmierer unter ProDOS. Mit der Diskette wird eine ausführliche deutsche Bedienungsanleitung mitgeliefert.
**Input 2.0** ..... 98,00
Bildschirm Masken Editor unter DOS 3.3 und ProDOS. Einbindbar in Applesoft- (auch kompilierte) und Assemblerprogramme. Diskette mit deutscher Bedienungsanleitung
**MMU 2.0** ..... 98,00
Memory Management Utilities für den Apple //e mit ext. 80 Z/Z-Karte. Über 25 Programme erschliessen neue Anwendungsmöglichkeiten.
**Softbreaker** ..... 48,00
Programm zur Herstellung von Sicherungskopien. Dieses Programm gestattet das softwaremäßige Unterbrechen, Speichern, Laden und Neustarten von geschützten Programmen. Für Apple//e mit 64kB-Erweiterung incl. dt. Bedienungsanleitung.
**Für obige Utilities steht eine ausführliche Beschreibung zur Verfügung. Bitte mit DIN A 5 - Freiumschlag anfordern.**

**A.C.E. Applesoft Command Editor** 149,00

ermöglicht einfaches Editieren einer BASIC-Zeile und bietet eine Menge Utilities: ermittelt ob eine Variable existiert, Auto Line Numbering, Monitor Kommandos können direkt von BASIC aufgerufen werden u. Keyboard Macros definiert werden.

**PASCAL-Systemhandbücher:**

**PASCAL Operating System Reference Manual** .. 45,00
**Satz-Language- und Operating Sys.Ref.Manual.** 75,00

**Software - Entwicklungshilfen**

Diese neue Serie für Softwareentwickler (=Workbench) werden als Einzelblattsammlung im Format DIN A 5 geliefert. Der passende Ordner bietet Platz für bis zu 3 Handbücher.
**DOS Programmers Toolkit (incl. Diskette)** ..... 279,00
Enthält: 6502 Assembler Toolkit mit vielen Hilfsprogrammen wie EDASM (Editor und Assembler), BUG BYTER (Programm zum Austesten von Assemblerprogrammen), RLOAD (Assemblerprogramme in Verbindung mit BASIC-Programmen).
THE APPLESOFT Programmers Toolkit enthält eine Sammlung von Hilfsroutinen für alle die in APPLESOFT programmieren.(Für Apple II+ und Apple IIe)
**ProDOS Technical Reference Manual (incl.Disk)** 95,00
Enthält ProDOS, das neue Betriebssystem von APPLE, sowie den EXERCISER (Interaktives Programm zum Austestenvon ProDOS durch Aufrufe von der Tastatur her). Das ProDOS Techn. Ref. Manual benutzt man am effektivsten in Verbindung mit dem ...
**ProDOS Assembler Tools ( incl. Diskette)** ..... 189,00
Enthält alles um effektiv unter ProDOS in Assembler zu programmieren. Beschreibung und Programme wie 6502 Assembler Tool Kit unter DOS 3.3.
**APPLE WORKBENCH Ordner (für ca. 3 Manuals)** . 37,00
Für 3-fach Lochung nach amerikanischer Norm.

**Ladenverkauf, Vorführung und fachgerechte Beratung**
**6900 Heidelberg 1 Breslauerstr. 29** Tel.06221/781500
**Geschäftszeiten:** Mo. - Fr. 9-13 +14 - 18 Sa. 9 -13 Uhr
**Versandanschrift:** 6900 Heidelberg 1 Dammweg 2

**IBM®-Gehäuse** ..... **229,-**

**ZUSATZ-KARTEN:**

**V-24-Schnittstelle** ..... **199,-**
**Z-80-Karte** ..... **139,-**
**80-Zeichen-Karte** m. Softswitch .... **236,-**
**16 K-Language-Karte** ..... **138,-**
**Centronics-Karte von Epson**
für Graphik .... **210,-** für Text .... **145,-**
**Eprommer** incl. Software ..... **198,-**

## Floppy-Controler

**FDC 4** für alle Laufwerke ..... **230,-**
Bausatz wie oben ..... **199,-**
Leerplatine wie ob. incl. Prom u. Eprom **130,-**
**Autopatch-Controler**
1 x 35 bis 2 x 80 Tr.-Disk, keine Patch-Disk notwendig, Gemischtbetrieb möglich . **298,-**
**auch für SONY 3½" Laufwerke**

**Joy Stick** ..... **69,-**
**Netzteil 5A** ..... **149,-**

**Monitore auf Anfrage**

**Halbschalen-Gehäuse** für 2 Slimline-Laufwerke ..... **65,-**
**Floppy-Kabel** 34pol. für 2 Laufwerke mit Shugart-Bus ..... **42,-**

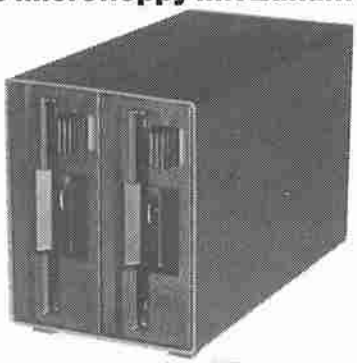
## Preh Commander Keyboards

Wir bieten Ihnen die **Preh-Qualität** auch für Apple. AK 88 spez. mit Gehäuse, Anschlußkabel, Zehner-Tastenfeld, dt. Zeichensatz, Sondertasten für Ctrl-Codes und Rechenfunktionen ..... **339,-**

**EPSON-Drucker FX 80** ..... **1670,-**
**EPSON-Drucker FX 100** ..... **2159.-**
**EPSON-Drucker RX 80** ..... **1079,-**
**EPSON-Drucker RX 80 FT** ..... **1295,-**

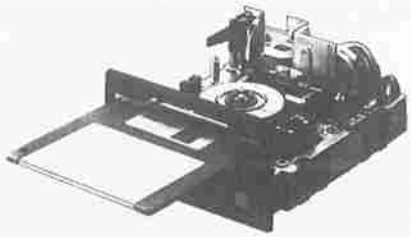
**Patch-Diskette** für **SONY 3½" Laufwerke** - ermöglicht die Anpassung an II/IIe und kompatible Computer ..... **80,-**
Manual vorab 15,-DM (wird beim Kauf der Patch-Diskette angerechnet).
**dto.** für 5¼" Laufwerke ..... **80,-**
**Disketten** Döbbelin&Böder SS/DD 10St. **62,-**
**10 Disketten** f. Sony 3½" Laufwerke . **150,-**

## Die Microfloppy mit Zukunft:



- Speicherkapazität: 2 x 1 MByte formatiert: 2 x 640 kByte
- Anschlußfertig mit PROM-residenter Patchsoftware für CP/M 2.2, Apple DOS 3.3 und Apple Pascal 1.1, Pro-DOS ab Januar 85

zum Preis von ..... **2499,-**



## SONY 3½" Laufwerk

...nur **849,-** ...deshalb gleich bestellen!

**Ueding electronics**

**Holtewiese 2** **DFÜ 02373/66877**
**5750 Menden 1** **Tel. 02373/63159**

## Apple ProDOS für Aufsteiger

Band 1

von Ulrich Stiehl
1984, 202 S., kart., DM 28,-
ISBN 3-7785-1027-4
Hüthig Verlag, Heidelberg

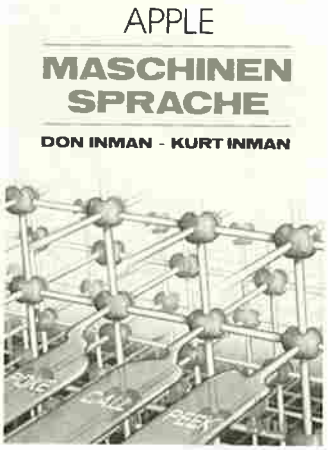
ProDOS ist das neue „professionelle DOS" (Professional Disk Operating System) für den Apple IIe sowie den mit einer Language Card ausgestatteten Apple II Plus. Reine Applesoft-Programmierer, die bereits unter DOS 3.3 programmiert haben, werden sich schneller an ProDOS gewöhnen, da die diesbezüglichen Unterschiede zwischen DOS 3.3 und ProDOS weniger gravierend sind. Sinngemäß liegt das Schwergewicht in dem ersten Band von „ProDOS für Aufsteiger" auf der Assembler-Programmierung, da Assembler-Programmierer unter ProDOS völlig umdenken müssen. Insbesondere sind alle früheren Assemblerprogramme unter ProDOS nicht mehr lauffähig und bedürfen einer intensiven Überarbeitung. Band 1 befaßt sich überwiegend mit den theoretischen Grundlagen von ProDOS, der internen und externen Speicherorganisation und enthält grundlegende Beispielprogramme für Assembler-Programmierer sowie generelle Untersuchungen zum BASIC-.SYSTEM. Da ProDOS über erheblich vielfältigere und leistungsfähigere, zugleich jedoch erheblich kompliziertere Dateistrukturen verfügt, sind theoretische Kenntnisse von ProDOS unabdingbar, wenn man die Features von ProDOS voll ausschöpfen will.

### Aus dem Inhalt:

Ein erster Überblick – ProDOS und DOS 3.3 – Interne Speicherorganisation – Externe Speicherorganisation – MLI (Machine Language Interface) – ProDOS für Applesoft-Programmierer

## Apple Maschinensprache

von Don und Kurt Inman
1984, 208 S.,zahlr. Abb. und Tabellen, DM 49,-
ISBN 3-921803-21-7
te-wi Verlag, München




Dieses Buch ist wahrscheinlich die beste Einführung in die 6502-Programmierung für denjenigen Assembler-Anfänger, der zuvor noch nie ein Maschinenprogramm geschrieben hat. Daß im ganzen Buch von Assemblern selbst nicht die Rede ist, hat einen didaktischen Hintergrund. Eine Reihe von 6502-Problemen, die dem Anfänger Schwierigkeiten bereiten, z.B. die bedingte Verzweigung und die damit zusammenhängenden Vorzeichenzahlen lassen sich auf der Maschinenebene wahrscheinlich besser verstehen, weil man hier gezwungen ist, die Sprungadresse selbst auszurechnen, was sonst der Assembler automatisch erledigt.

### Aus dem Inhalt:

Applesoft II BASIC - kurzgefaßt – Alles über Zeichen – Alles über Speicher – Alles über Maschinenbefehle – Maschinenprogramme mit BASIC eingeben – Graphik – Text – Ton – Arithmetik – Was tun mit den Maschinenprogrammen

## Apple Assembler

Tips und Tricks

von Ulrich Stiehl
1984, 226 S., 3 Abb., kart., DM 34,-
ISBN 3-7785-1047-9
Hüthig Verlag, Heidelberg

„Apple Assembler" wendet sich an alle, die bereits Anfängerkenntnisse der 6502-Programmierung haben – z.B. aufgrund des Buches „Apple Maschinensprache" – und nunmehr ein Nachschlagewerk für ihren Apple II Plus/IIe/IIc suchen, in dem alle wichtigen ROM-Routinen sowie eine Vielzahl sonstiger Hilfsprogramme in einer systematischen Form zusammengestellt werden. Insgesamt umfaßt dieses Buch über 40 Utilities, darunter mehrere völlig neuartige Programme wie Double-Lores, Double Hires, Screen-Format u.a.
Der erste Teil enthält ein Repetitorium der wichtigsten Befehle, Adressierungsarten und sonstigen Besonderheiten des 6502.
Im zweiten Teil werden alle Adressen des Monitors zusammengestellt, die für Assembler-Programmierer von Nutzen sein können. Darüber hinaus findet der Leser Unterroutinen für hexadezimale Addition/Subtraktion/Multiplikation/Division, Binär-Hex-ASCII-Umwandlung usw.
Der dritte Teil befaßt sich mit der Speicherverwaltung der Language Card und der IIe-64K-Karte und enthält Move-Programme zum Verschieben von Daten in die und aus der Language Card sowie der 64K-Karte.

Der vierte Teil ist dem Applesoft-ROM gewidmet und listet eine große Anzahl nützlicher Interpreter-Adressen. Bei den Utility-Programmen liegt das Schwergewicht auf Fließkommamathematik einschließlich Print Using.
Der letzte Teil behandelt den Text- und Graphikspeicher. Neben einem professionellen Maskengeneratorprogramm werden auch Routinen zur Double-Lores- und Double-Hires-Grafik vorgestellt.

### Aus dem Inhalt:

6520-Repetitorium – Monitor-ROM – Speicherverwaltung – Applesoft-ROM – Text- und Grafikspeicher

## Pandasoft-Katalog

Soeben ist der neue Katalog Winter 1984 der Firma Pandasoft Dr. Eden, Berlin, erschienen. Auf 86 Seiten findet der Apple-Benutzer eine sehr große Auswahl an Programmen und Zusatzkarten zu günstigen Versandhauspreisen. Die Hälfte des Katalogs nehmen inzwischen Bücher und sonstige Begleitliteratur zum Apple ein.
Bezugsquelle: Pandasoft, Uhlandstr. 195, 1000 Berlin 12, Tel. 030/310423

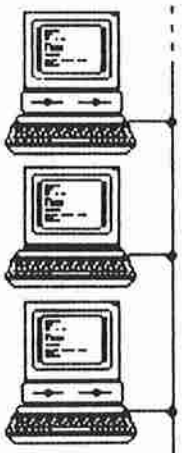
# Die Massenspeicher-Profis für Apple Computer

**wir haben die derzeit leistungsstärksten Subsysteme !**

**→ Apple II, III + Kompatible:**

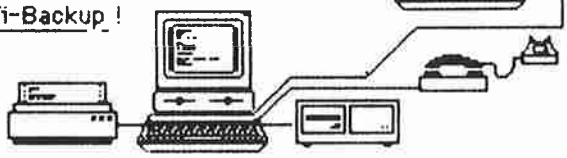
- Hard Disk/Festplatten-Subsysteme
- Wechselplatten-Subsysteme
- Fest-/Wechsel-Subsysteme
- Bandstreamer für Backup
- NC-Net Lokales Netzwerk
- NCW 2.0 Multibetriebssystem-Software
- 8" Floppy mit SS/SD + DS/DD, IBM-komp.
- i.V.: Fest-/Wechselplatten für Macintosh und Apple IIc, APPLE-BUS Kompatibilität
- Schon für die Einzelplatz Fest/Wechselplatten gelten konkurrenzlose Leistungen...

... bis zu 63 Keywords/Benutzer definier- und editierbar. Wahlfrei unter DOS 3.3 (inkl. Varianten), ProDOS, UCSD-Pascal 1.1 u. 1.2, CP/M sowie NCNDOS (Multiuser-DOS) einsetzbar. Betriebsystem-Wechsel per Befehl in Sekunden. Für jedes Keyword komplette Konfigurations-Umgebung in allen Betriebsystemen definierbar. Schreib/Leseschutz f. alle Keywords u. "logische Unit-Bezeichnungen" definierbar. Änderung der Plattenaufteilung ohne (!) Datenverlust in allen Betriebsystemen. Turnkey-Lösungen .. und .. und ... Profi-Backup !



**NovoComp** Datensysteme GmbH
**Walramsneustraße 7 + 9**
**D-5500 Trier**
**(0651) 4 22 44** Tlx. 472569

US-Fachleute bewerten:
"Die mit Abstand beste Lösung am Markt!"



## D.O.S. Computersysteme

**Der Spezialist für Schulen, Universitäten, Techniker!**

**Aus unserem umfangreichen Angebot:**

| | |
|---|---|
| Profimax II mit 64K RAM, Apple II-comp., 6502 + Z80, progr. Tastatur num. Block, 5A-Netzteil | 1248,– |
| Profimax III, wie oben, im IBM-Gehäuse mit IBM-Tastatur | 1448,– |
| Motherboard Profimax | 798,– |
| Profimonitor, Metallgehäuse, 22 MHZ, 12", bernstein | 398,– |
| Monitor Heath/Zenith ZVM 123, grün, 15 MHZ | 308,– |
| Tastatur aus Profimax II | 198,– |
| IBM-Tastatur, ASCII, anschlußfertig an Apple o. - comp. | 348,– |
| IBM-Tastatur, deutsche Belegung | 398,– |
| TEAC 55A, kompl. m. Kabel u. Geh., anschlußfertig an Apple-Contr. | 648,– |
| TEAC 55F, 2 × 80 Spuren, modif. für Betrieb an Apple, Speicherriese! | 798,– |
| Winchester-Laufwerk, 10 MB,Geh., Netzt., Kabel, Software, Contr. | 4998,– |
| Disk II Controller | 148,– |
| Contr. für alle 5 1/4" Laufwerke, mit Patch für DOS, CP/M, PASCAL | 248,– |
| Z80-Softcard | 148,– |
| 80Z-Karte, Softswitch, 2 Zeichens. | 268,– |
| 80Z-Karte mit 4 Zeichensätzen | 268,– |
| Druckerinterface, par. (= Centr.), für NEC 8023, ITOH 8510, grafikf. | 298,– |
| Epromburner für 2716-2764 | 198,– |
| AD-Wandler, 333ms Wandelz., 1 Kanal | 288,– |
| Drucker NEC 8023B/N (= ITOH 8510) | 1498,– |
| Typenraddrucker JUKI 6100 !!!! | 1898,– |
| Typenradschreibm. Olympia electr. compact 2, Centronics-Schnittst. | 1498,– |
| Plotter Pixy 3, DIN A 4 | 2498,– |
| Disks Scotch, SS, DD, 10 St. | 70,– |

Neu bei uns: HP-Rechner zum Sonderpreis. Profimax PC, der IBM-Kompatible mit DOS 2.11. – jetzt lieferbar.

Wir sind nicht billig – wir sind preiswert!

Fordern Sie unseren ausführlichen Katalog an (DM 2,– in Briefmarken).

**D.O.S. Computersysteme**
**Martin Götzer**
Am Kühnbach 42 · 7170 Schwäbisch Hall 11
Telefon 0791/51736
Urlaub vom 22.12.84 bis 13.1.85

## Das ist Spitze!

| | |
|---|---|
| Apple IIe 64 KB | DM 2.340,– |
| Apple Disk II m.C. | DM 1.155,– |
| Apple Disk II 2. LW | DM 850,– |
| Duo Disk 2 × 143 KB m.C. | DM 2.000,– |
| Apple II Monitor | DM 552,– |
| Matrixdrucker Itoh 8510 A | DM 1.500,– |
| Imagewriter 80 Z incl. Kit | DM 1.480,– |
| 80 Zeichen / 64 KB IIe | DM 783,– |
| Z-80 Card | DM 140,– |
| 16 K-Karte | DM 140,– |
| Akustikkoppler m. FTZ | DM 695,– |
| Macintosh | DM 6.350,– |
| Macintosh 2. LW | DM 1.265,– |
| 2.000 Bl. TAB - papier 240 × 12" weiß, perforier | DM 47,– |
| Abdeckhaube IIe | DM 48,– |
| Abdeckhaube Mac | DM 48,– |
| Abdeckhaube Itoh 8510 A | DM 21,– |
| Apple IIc 128 KB | DM 2.800,– |
| Apple IIc 2. LW | DM 820,– |
| Apple IIc Monitor | DM 495,– |
| Apple IIc Ständer | DM 98,– |
| Apple Writer IIe | DM 490,– |
| Lagerhaltung | DM 550,– |
| Kundenkartei | DM 450,– |

Alle Preise verstehen sich zuzügl. MwSt.

**ORGASOFT® GMBH**
Organisationsberatung + Software

**Organisationsberatung + Software**
**Microcomputer**

Werner-v.-Siemens-Straße 3
7730 VS-Villingen
Telefon 07721/72213 + 72223

Computer Ring
Der Btx.-Fachmann in ihrer Nähe

## Akustik-Koppler-Set mit FTZ-Nr.

| Für C 64 | Für Apple II |
|---|---|
| Koppler | Koppler |
| Diskette | Diskette |
| Kabel | Kabel |
| Stecker | Stecker |
| anschlußfertig | anschlußfertig |
| Handbuch | Handbuch |
| Best.-Nr. 8560 | Best.-Nr. 8561 |
| **DM 795,–** | **DM 795,–** |

**TONACORD**
2330 Eckernförde, Sauerstr. 13
Tel. (04351) 4039
Telex 17435130

## EDV-Programme für Apple-Computer

1. **Finanzbuchhaltung**
   (einschließlich Kunden und Lieferanten, offener Posten, Umsatzsteuervoranmeldung usw.)
2. **Baufinanzierung**
   (alle geltenden Gesetze auf dem neuesten Stand und im Handbuch ausführlich erläutert)
3. **Adressen- und Textverwaltung**
   (Erstellung von Serienbriefen mit individueller Anrede, Auswahl nach verschiedenen Kriterien usw.)

Ausführliche Handbücher
Information,
Lizenzerteilung,
Schulung:

CPS computer

**CPS–DATENSERVICE GMBH**
**Frankfurter Str. 126, 6050 Offenbach**
**Telefon 069–88 05 90**

Mit 15 Seiten Einkommensteuer Professionelles Basic-Programm

# peeker

## Vorschau Heft 1/2-85*

## DOS 3.3

**Poor Man's RAM-Disk**
RAM-Disk-Driver für die Language Card

**Die RAM-Karten von IBS**
Mit Quellcode eines RAM-Disk-Drivers

## ProDOS

**Quickcopy**
Ein schnelles ProDOS-Kopierprogramm für 35-80 Spuren

**ProDOS und „Kompatible"**
Ein Nachtrag

## Utilities

**Superschnelle Garbage Collection**

**Speichern Sie den Bildschirm ab!**
Eine Utility für 40- und 80-Z/Z-Bildschirm

## Applesoft

**Applesoft-Erweiterungen ganz einfach**
Die CHRGET-Manipulation

**Testgenerator für Legastheniker**

**Macros für PRODOS.EDITOR**

## Pascal

**Pascal-Directory näher beleuchtet**

**Konvertierung von Pascal- in DOS-Textfiles**

**Konvertierung von DOS- in Pascal-Textfiles**

## Macintosh

**Ikonen und Deixis**
oder die Philosophie des Macintosh

**Microsoft Basic leicht geMACht**
Teil 2

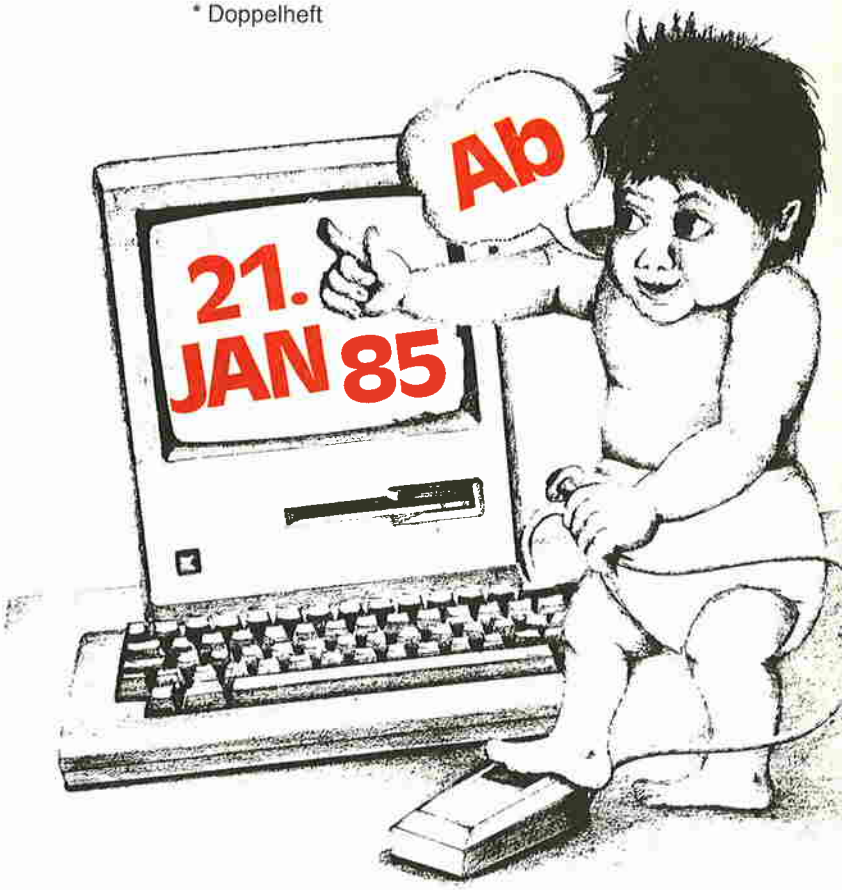
## Hardware-Test

**Weltweite Datenübertragung mit dem WS 2000**
Ein Universalmodem für alle Normen

**Exbasic Level im Test**

**Masterclock von Hoco**

* Doppelheft



# Sie haben einen APPLE ...

## Wir liefern ALLES Drumherum

## Software Hardware Literatur

## Fordern Sie unseren Gratiskatalog an!

**Alles für den APPLE II+, IIe, IIc & Macintosh**

**pandasoft** Dr.-Ing. Eden

Uhlandstraße 195 (Am Steinplatz), 1000 Berlin 12

Mo.–Fr. 10–18, Sa. 10–13 Uhr · ☎ 030/310 423 · Telex: 185 859

Ich besitze einen APPLE.
Bitte schicken Sie mir Ihren kostenlosen Gratiskatalog

Name: .....

Adresse: .....

.....

# PRO METRIC® IST DA!

| | |
|---|---|
| **B2-Motherboard mit 6502, 65C02C, Z80B, V24, Parallel-Schnittstelle, 80-Zeichenkarte, RGB-Ausgang, 192 KB RAM** | **DM 2299,--** |
| **B3-Motherboard, wie B2, zusätzlich 256 KB Pseudofloppy** | **DM 2899,--** |
| **Prometric® B2 Tischgerät** | **ab DM 3349,--** |
| **Prometric® B3 Tischgerät** | **ab DM 3999,--** |
| **Prometric® Portable B2, Monitor 9"** | **ab DM 4449,--** |
| **Prometric® Portable B3, Monitor 9"** | **ab DM 4999,--** |

**Für feste Bestellungen, die bis zum 15. Januar 1985 bei uns eingehen, gewähren wir einen Einführungsrabatt von 10%!!!!!!!**

**Neu von DRV:** C-DOS, das 80-Track DOS für B2 und Kompatible, verarbeitet Integer, DOS 3.3 und Prodos-Files DM 149,--

DEUTSCHES BASIC, ROM-residient, übersetzt Ihre Apple-Soft Files ins Deutsche und umgekehrt DM 149,--

CALVADOS, deutsche Textverarbeitung DM 299,--

**Wir suchen Stützpunkt-Händler mit entsprechendem finanziellen Hintergrund. Anfragen bitte nur schriftlich.**

## Welt-Neuheit

**Code-Breaker, kopiert alle indexsynchronisierten Disketten.**
**Wir suchen Firmen für die Herstellung und den Vertrieb.**

| | | | |
|---|---|---|---|
| TEAC FD55F | DM 684,-- | STAR-DRUCKER | a. Anfr. |
| FDC4-Controler | DM 185,-- | **YE-DATA 480** | **DM 694,--** |
| Shugart-Bus-Kabel | DM 45,-- | ERPHI-Controler | DM 298,-- |
| PREH-Commander AK87 | DM 329,-- | IBM-look Gehäuse | DM 218,-- |
| MONITOR 12", 22 MHZ | DM 349,-- | MONITOR 15" | DM 528,-- |
| M80-DRUCKER | DM 748,-- | JUKI 6100 | DM 1748,-- |
| Graphik-Par.-Interface | DM 125,-- | 32K-PRINTER-BUFFER | DM 389,-- |
| PSEUDO-FLOPPY 256 KB | DM 899,-- | MOTHERBOARD 64 KB + Z80 | DM 699,-- |

Alle Preise inklusive 14% MwSt. Weiteres Zubehör auf Anfrage.

**E. Böhmer, DRV, Am Kellersbusch, 6072 Dreieich, 0 61 03 / 8 46 47**